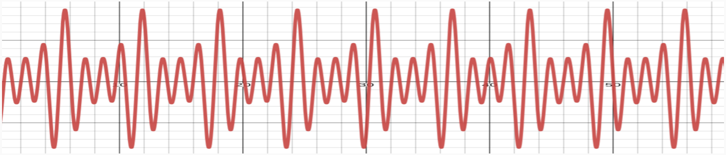
Regression Analysis



Prof. Fabrício Olivetti de França

Federal University of ABC

05 February, 2024

Regression

Regression

Regression: a return to a previous and less advanced or worse state, condition, or way of behaving.

Regression to the mean

Francis Galton observed that extreme traits in parents are not completely passed to their offsprings.

A child from tall parents is taller than average but not as tall as their parents.

Likewise, a child from short parents is shorter than average but not as short as their parents.

Regression to the mean

They both regressed towards the mean!

Regression to the mean

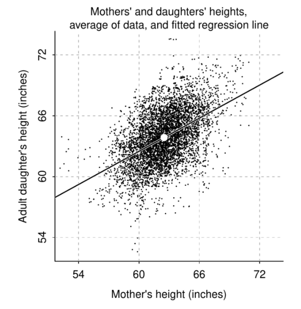


Figure 1: Source: Gelman, Andrew, Jennifer Hill, and Aki Vehtari. Regression and other stories. Cambridge University Press, 2020.

We can fit this data with

$$y = 30 + 0.54x$$

where y is the height of the daughter and x is the height of the mother, in inches.

Regression to the mean

For a better interpretation, we can rewritte it as:

$$y = 63.9 + 0.54(x - 62.5)$$

When a mother has an average height (62.5), her daughter is predicted to have 63.9 in her adulthood.

Regression to the mean

$$y = 63.9 + 0.54((x+1) - 62.5) = 63.9 + 0.54(x - 62.5) + 0.54$$

For every inch above average, the daughter is expected to be half an inch taller.

Regression to the mean

$$y = 63.9 + 0.54((x - 1) - 62.5) = 63.9 + 0.54(x - 62.5) - 0.54$$

For every inch below average, the daughter is expected to be half an inch shorter.

Regression to the mean

So the height of the daughter depends on how much the mother’s height deviates from the mean.

Linear Regression

Linear Regression returns estimates for the outcome using the model:

$$y^{(i)} = \sum_{j=1}^{P} \beta_j x_j^{(i)} + \epsilon^{(i)}$$

where β_j is the j-th parameter and $\epsilon^{(i)}$ is a measurement disturbance of the i-th example.

Linear Regression

This regression model is composed of a **deterministic** part, depending on β, x, and a stochastic part, depending on ϵ.

Linear Regression

In matrix notation, we can write the model for N cases and P parameters as:

$$y = x\beta + \epsilon$$

where $y, \epsilon \in \mathbb{R}^{N \times 1}, \beta \in \mathbb{R}^{P \times 1}, x \in \mathbb{R}^{N \times P}$.

Linear Regression

We can assume a zero mean for ϵ which leads us to:

$$
\begin{aligned}
E[y] &= E[X\beta + \epsilon] \\
E[\epsilon] &= 0 \\
E[y] &= E[x\beta]
\end{aligned}
$$

$x\beta$ is also called the **expectation function** of the regression model.

Normality

It is a common assumption for linear regression that ϵ is normally distributed:

$$Var[\epsilon] = E[(\epsilon - E[\epsilon])(\epsilon - E[\epsilon])^T]$$
$$Var[\epsilon] = E[\epsilon\epsilon^T] = \sigma^2 I$$

where the superscript T means transpose, and I is the identity matrix of size $N \times N$.

Normality

Remembering the probability density function:

$$f(x \mid \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Normality

The joint probability function of our model $p(y \mid \beta, \sigma^2)$ can be calculated as:

$$\begin{aligned}
p(y \mid \beta, \sigma^2) &= \prod_{i=1}^{N} f(y^{(i)} \mid \beta, \sigma^2) \\
&= \frac{1}{(\sigma\sqrt{2\pi})^N} e^{-\frac{1}{2}\left(\frac{(y-x\beta)(y-x\beta)^T}{\sigma}\right)^2} \\
&= (2\pi\sigma^2)^{-N/2} e^{-\frac{1}{2}\left(\frac{(y-x\beta)(y-x\beta)^T}{\sigma}\right)^2}
\end{aligned}$$

Likelihood function

The likelihood function $l(\beta, \sigma \mid y)$ is the probability of observing y if β, σ were the true values.

In functional form $l(\beta, \sigma \mid y) = p(y \mid \beta, \sigma^2)$.

Likelihood function

We want to find the values of β that maximizes the likelihood function. As the most likely β for the observed values of y.

This is the best fit for our data, given the assumptions made so far.

Likelihood function

We can transform the likelihood with the log function to make this optimization simpler. The **negative log-likelihood** (nll) can be written as:

$$
\begin{aligned}
nll(\beta, \sigma \mid y) &= -\log\left(l(\beta, \sigma \mid y)\right) \\
&= -\log\left(\prod_{i=1}^{N} f(y^{(i)} \mid \beta, \sigma^2)\right) \\
&= -\log\left((2\pi\sigma^2)^{-N/2} e^{-\frac{1}{2}\left(\frac{(y-x\beta)(y-x\beta)^T}{\sigma}\right)^2}\right) \\
&= -\log\left((2\pi\sigma^2)^{-N/2}\right) + \frac{1}{2}\left(\frac{(y-x\beta)(y-x\beta)^T}{\sigma}\right)^2 \\
&= \frac{N}{2}\log\left((2\pi\sigma^2)\right) + \frac{1}{2\sigma^2}\left((y-x\beta)(y-x\beta)^T\right)^2
\end{aligned}
$$

Likelihood function

Since we are interested in finding the optimal β, we can discard the constant terms and σ:

$$nll(\beta, \sigma \mid y) = \frac{1}{2}\left((y - x\beta)(y - x\beta)^T\right)^2$$

and minimize this function. We call $\hat{\beta}$ the **maximum likelihood estimate**.

Likelihood function

The maximum likelihood estimate for the linear regression is given by:

$$\hat{\beta} = (x^T x)^{-1} x^T y$$

This is called the **least squares estimate**.

Least Squares

The least squares is appropriate when:

- The expectation function is correct
- The response is the expectation plus a disturbance
- The disturbance is independent from the expectation
- The disturbance of a point is normally distributed with zero mean
- The disturbances are independetly distributed with equal variance

Point Prediction

With a $\hat{\beta}$ we can make a point estimate for the i-th data point as $\hat{y}^{(i)} = \hat{\beta}x^{(i)}$.

In the same way, $\hat{\beta}$ is the point estimation of the parameters.

Is x_i really important?

One part of regression analysis is concerned whether a predictor is important to the model.

For linear regression this is basically asking: is $\beta = 0$?

Statistical test

For this purpose we can perform an statistical test. A common setting for statistical test for regression is to pose a **null hypothesis**:

$$H_0 : \beta_i = 0$$

Statistical test

Similarly we define the alternative hypothesis:

$$H_a : \beta_i \neq 0$$

Statistical test

For this particular setting, since ϵ is normally distributed, and Y is a linear function of ϵ, it follows that β is also normally distributed.

But when we work only with an esimate $\hat{\beta}$ and, thus, the variance is also estimated, we assume a t-distribution.

Student’s t Distribution

Similar to a Normal distribution but with heavier tails. Allows us to compensate the uncertainties by the degrees-of-freedom.

$$f(x \mid \nu, \mu, \sigma) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{\pi\nu\sigma^2}} \left(1 + \frac{(x-\mu)^2}{\nu\sigma^2}\right)^{-\frac{\nu+1}{2}}, \qquad (1)$$

Student’s t Distribution

$$\begin{aligned} E[f(x \mid \mu, \sigma)] &= \mu \\ Var[f(x \mid \mu, \sigma)] &= \frac{\nu}{\nu - 2}\sigma^2 \nu > 2 \end{aligned}$$

Student's t Distribution

$$\nu = 2, \mu = 0, \sigma = 0.2$$

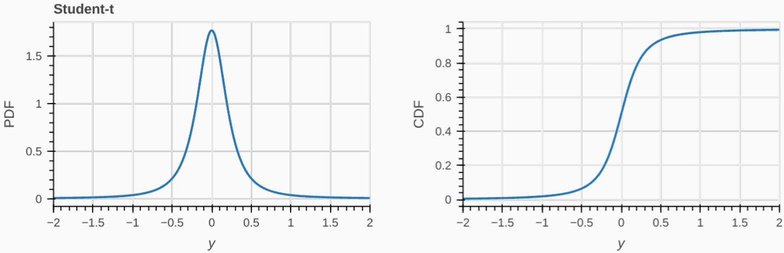


Figure 2: https://distribution-explorer.github.io/continuous/student_t.html

hypothesis test

So, the hypothesis test is calculated from the t-Statistic:

$$t = \frac{\text{estimated value} - \text{hypothesis}}{\text{standard error of estimate}}$$

Properties of $\hat{\beta}$

Since β is normally distributed, we have:

$$\begin{aligned}
E[\hat{\beta}] &= \beta \\
Var[\hat{\beta}] &= E[\hat{\beta}^2] - E[\hat{\beta}]^2 \\
&= E[\Big((x^Tx)^{-1}x^Ty\Big)^2] - \beta^2 \\
&= E[\Big((x^Tx)^{-1}x^T(x\beta+\epsilon)\Big)^2] - \beta^2 \\
&= E[\Big((x^Tx)^{-1}x^Tx\beta+(x^Tx)^{-1}x^T\epsilon\Big)^2] - \beta^2 \\
&= E[\Big(\beta+(x^Tx)^{-1}x^T\epsilon\Big)^2] - \beta^2 \\
&= E[\beta^2] + 2(x^Tx)^{-1}x^T\beta E[\epsilon] + E[((x^Tx)^{-1}x^T\epsilon)^2] - \beta^2 \\
&= E[((x^Tx)^{-1}x^T\epsilon)^2]
\end{aligned}$$

Properties of $\hat{\beta}$

cont.

$$
\begin{aligned}
&= ((x^T x)^{-1} x^T)^2 E[\epsilon^2] \\
&= (x^T x)^{-1} x^T (x^T x)^{-1} x^T \sigma^2 \\
&= (x^T x)^{-1} I \sigma^2 \\
&= \sigma^2 (x^T x)^{-1}
\end{aligned}
$$

Note that $Var[\hat{\beta}]$ is a matrix where the diagonals are the individual variance of each estimate $\hat{\beta}_i$ and every element i, j is the **covariance** of $\hat{\beta}_i$ and $\hat{\beta}_j$.

$E[\hat{\beta}]$ says that the estimation is unbiased, $Var[\hat{\beta}]$ says that the covariance of the estimates depends on the variance of ϵ and the data points matrix x.

Standard error

The standard error of $\hat{\beta}$ is:

$$se(\hat{\beta}) = \sqrt{s^2 \operatorname{diag}((x^T x)^{-1})}$$

$$s^2(\hat{\beta}) = \frac{1}{N-P} \left((y - x\beta)(y - x\beta)^T \right)^2$$

where diag is the diagonal of the matrix and s^2 is the estimated variance with $N - P$ degrees of freedom since we don’t know the true σ^2.

hypothesis test

We then establish a rejection threshold for our hypothesis test. Let us use $\alpha = 0.05$.

$$t = \frac{\hat{\beta}_i - 0}{\mathrm{se}(\hat{\beta}_i)}$$

From the t-Statistic we can calculate a p-value and, if the p-value is less than α, we reject the hypothesis that $\beta_i = 0$.

hypothesis test

If we assume that $\beta_i = 0$ and we obtain $p = 0.05$ for our current sample, it means that if we take many different samples and repeat our experiment, we would observe the obtained value of $\hat{\beta}_i$ is 5% of them due to random error.

As the chance of obtaining such value due to error is too small, we reject the hypothesis.

Confidence Intervals

So far we have been working with the point estimates of our predictions and parameters.

But since there are uncertainties involved as we are estimating the parameters with a sample of the whole population, it is more insightful if we have an interval of possible values for these parameters and for the predictions.

Confidence Intervals

The **confidence interval** (CI) is another way to view the hypothesis test.

We say that the $(1 - \alpha)100\%$ of an estimate $\hat{\beta}$ is the range of values that would reject the null hypothesis when using α.

Marginal Confidence Interval

We calculate the $(1-\alpha)100\%$ marginal confidence interval of $\hat{\beta}$ is:

$$\frac{\hat{\beta}}{se(\hat{\beta})} \leq |t_{(\alpha/2,\nu)}|$$

$$\hat{\beta} \pm t_{(\alpha/2,\nu)} se(\hat{\beta})$$

where $\nu = N - P$ degrees of freedom, $s(\hat{\beta})$ is the standard error of β and $t_{(\alpha/2,\nu)}$ is the quantile function at $\alpha/2$ for the Student's t distribution with ν degree of freedoms.

This is the CI for an individual $\hat{\beta}_i$.

Interpreting the CI

ℹ A $(1-\alpha)100\%$ CI can be interpreted as the range of values in which contains the true parameter with $(1-\alpha)100\%$ certainty.

Joint Confidence Interval

Alternatively, we may test the hypothesis that all of the parameters are equal to 0. In this situation, we cannot simply calculate the individual hypothesis as the repeated test would increase the chance of type rejecting a valid null hypothesis.

The probability of making one error in m tests is $1 - (1 - \alpha)^m$, for $m = 5$ we reach approx. 23% of chance!

Joint Confidence Interval

For multiple regression, we can use the Fisher’s F distribution with P and $N - P$ degrees of freedom.

Fisher’s F Distribution

The F Distribution is the ratio of X^2 distributions of two populations with distinct variances.

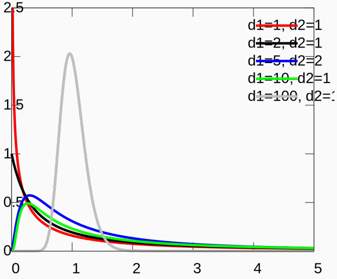


Figure 3: https://en.wikipedia.org/wiki/F-distribution

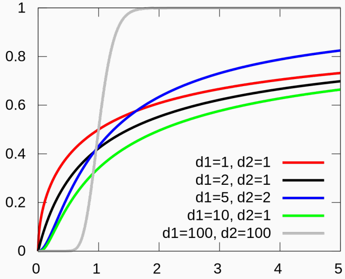


Figure 4: https://en.wikipedia.org/wiki/F-distribution

Joint Confidence Interval

We can calculate the joint confidence interval as the ellipsoid:

$$(\beta - \hat{\beta})^T x^T x(\beta - \hat{\beta}) \leq Ps^2 F(\alpha, P, N - P)$$

where $F(\alpha, P, N - P)$ is the quantile for the Fisher's F distribution with P and $N - P$ degrees of freedom.

Joint Confidence Interval

The inequation of the joint distribution describes an ellipsoid, for example:

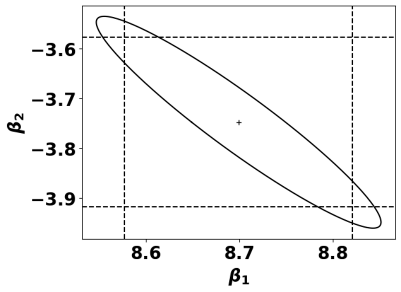


Figure 5: Reproduced from: Bates, Douglas. “Nonlinear regression analysis and its applications.” Wiley Series in Probability and Statistics (1988).

Confidence Interval of a Prediction

Likewise, we can calculate a confidence interval of an expected value at $x^{(i)}$:

$$(x^{(i)})^T \hat{\beta} \pm s\sqrt{(x^{(i)})^T (x^T x)^{-1} x^{(i)}}\, t(\alpha/2, N - P)$$

And a confidence band with:

$$x^T \hat{\beta} \pm s\sqrt{x^T (x^T x)^{-1} x}\sqrt{PF(\alpha, P, N - P)}$$

Calculating the Least Squares estimate

To calculate the least squares estimate, we have to find the inverse of $x^T x$:

$$\boxed{\hat{\beta} = (x^T x)^{-1} x^T Y}$$

It is easier to calculate it using a matrix decomposition technique.

Calculating the Least Squares estimate

For example, we can use the QR decomposition that decomposes a matrix x into

$$
\begin{aligned}
x &= qr \\
q^T q &= q q^T = I \\
r &= \begin{bmatrix} 1 \\ 0 \end{bmatrix}
\end{aligned}
$$

where q is orthogonal and r is zero below the main diagonal with the upper diagonal having size $P \times P$

Calculating the Least Squares estimate

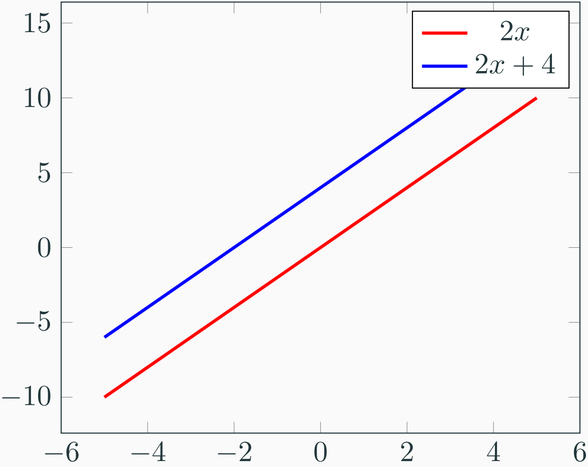
The value of $\hat{\beta}$ can be calculated using the relation:

$$r_1\hat{\beta} = q_1^T y$$

where q_1 is the first p columns of q and, since r_1 is upper diagonal, we can find the estimates with back substitution.

Some practical considerations

It is common practice to have an additional column in the predictors with all elements equal to 1 as an intercept to account for the average value when the predictors are all 0.



Example: grades

Let’s create the model $y = \beta_1 + \beta_2 \, \text{ETA\_mean}$.

```python
df = pd.read_csv("grade.csv")
xcols = ['ETA_mean']

x, y = df[xcols].values, df.grade.values
x = np.vstack([np.ones(x.shape[0]), x[:,0]]).T
```

Example: grades

```python
n = y.shape[0]
p = x.shape[1]
dof = n - p
reg = LinearRegression(fit_intercept=False)
reg.fit(x, y)

y_hat = x @ betas_qr
s2 = np.sum((y - y_hat)**2)/dof
se = np.sqrt(s2 * np.diag(np.linalg.inv(x.T @ x)))

print("parameters: ", reg.coef_)
print("R^2 score: ", reg.score(x, y))
print("degree of freedom: ", dof, "\ns2: ", s2, "\nstandard errors: ",
```

Example: grades

```
parameters:  [ 8.6992847  -3.74766658]
R^2 score:  0.1890555566476365
degree of freedom:  7998
s2:  3.3845582006552553
standard errors:  [0.06239017 0.08679028]
```

Example: grades

```python
# Let's try QR and Cholesky decomposition
p = x.shape[1]
q, r = sc.linalg.qr(x)
q1 = q[:,:p]
r1 = r[:p, :p]
betas_qr = sc.linalg.solve_triangular(r1, q1.T@y)

l = sc.linalg.cho_factor(x.T @ x)
inv = sc.linalg.cho_solve(l, np.identity(p))
betas_chol = inv @ x.T @ y

print("QR: ", betas_qr)
print("Cholesky: ", betas_chol)
print("scikit-learn: ", reg.coef_)
```

Example: grades

```
QR:  [ 8.6992847  -3.74766658]
Cholesky:  [ 8.6992847  -3.74766658]
scikit-learn:  [ 8.6992847  -3.74766658]
```

Example: grades

```python
alpha = 0.05
marginal_betas = []
for i, coef in enumerate(betas_qr):
    v = np.abs(se[i]*t.ppf(1 - alpha/2, n-p))
    marginal_betas.append((coef - v, coef + v))
    print(f"{coef - v:0.2f} <= beta_{i} <= {coef + v:0.2}")
```

Example: grades

```
8.58 <= beta_0 <= 8.8
-3.92 <= beta_1 <= -3.6
```

Example: grades

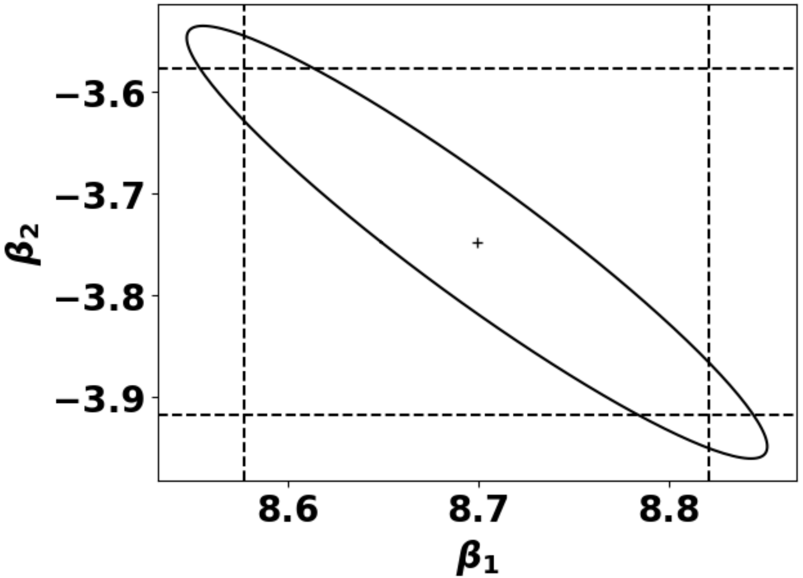
```
# joint interval
omegas = np.pi*np.arange(0, 2.01, 0.01)
const = np.sqrt(p*s2*f.ppf(1-alpha, p, n-p))
            *np.linalg.inv(r1)
betapts = np.array([betas_qr + const
              @ np.array([np.cos(w), np.sin(w)])
                  for w in omegas])
```

Example: grades

```python
_,ax = plt.subplots()
ax.plot(betapts[:,0], betapts[:,1], color='black')
ax.plot(betas_qr[0], betas_qr[1], '+', color='black')
ax.axvline(x=marginal_betas[0][0], linestyle='--', color='black')
ax.axvline(x=marginal_betas[0][1], linestyle='--', color='black')
ax.axhline(y=marginal_betas[1][0], linestyle='--', color='black')
ax.axhline(y=marginal_betas[1][1], linestyle='--', color='black')

print(np.min(betapts[:,0]))
ax.set_xlabel(r"$\beta_1$")
ax.set_ylabel(r"$\beta_2$")
```

Example: grades



Example: grades

```
for i, y_i in enumerate(reg.predict(x)):
  c = np.abs(t.ppf(1 - alpha/2, dof))
  v = np.sqrt(s2 * (x[i,:].T @ np.linalg.inv(x.T @ x) @ x[i,:]))*c
  print(f"{y_i - v:0.2f} <= y_{i} <= {y_i + v:0.2f}")
  if i > 10:
    break
```

Example: grades

```
6.41 <= y_0 <= 6.50
6.40 <= y_1 <= 6.48
5.50 <= y_2 <= 5.60
4.85 <= y_3 <= 4.98
5.26 <= y_4 <= 5.37
5.28 <= y_5 <= 5.39
5.73 <= y_6 <= 5.82
6.35 <= y_7 <= 6.43
6.29 <= y_8 <= 6.38
4.80 <= y_9 <= 4.94
4.46 <= y_10 <= 4.62
5.06 <= y_11 <= 5.18
```

Example: grades

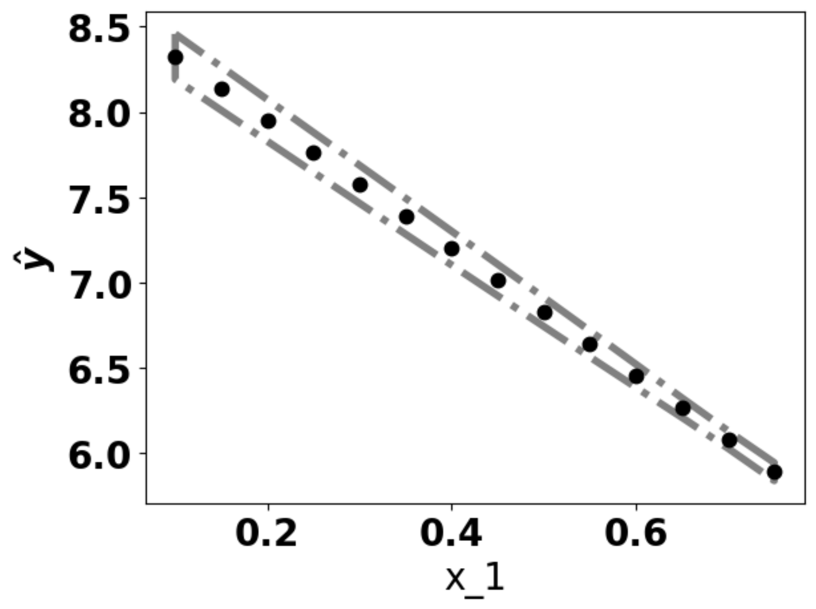
```python
xi = np.arange(0.1, 0.8, 0.05)
xi = np.vstack([np.ones(xi.shape[0]), xi]).T
c = np.sqrt(s2*p * f.ppf(1-alpha, p, n-p))
xx = np.sqrt(xi @ np.linalg.inv(x.T @ x) @ xi.T)

yy = np.max(np.abs(xx * c), axis=1)
y_h = xi @ betas_qr
```

Example: grades

```python
_,ax = plt.subplots()
ax.plot(xi[:,1], y_h, '.', color='black', markersize=15)
ax.fill_between(xi[:,1], y_h - yy, y_h + yy, edgecolor='gray',
    facecolor='white', alpha=1,
    linewidth=4, linestyle='dashdot', antialiased=True)
ax.set_xlabel('x_1')
ax.set_ylabel(r'$\hat{y}$')
```

Example: grades



Example code

Cholesky decomposition of an array x of size $m \times m$:

```
def chol(x):
  m = x.shape[0]
  chol = np.zeros((m, m))
  for j in range(m):
    for i in range(j,m):
      delta = x[i,j] - np.dot(chol[i,:j], chol[j,:j])
      chol[i,j] = np.sqrt(delta) if i==j
                                 else delta/chol[j,j]
  return chol
```

Example code

Inversion of x using Cholesky:

```
def inv(x):
  l = chol(x)
  tmp = l.copy()
  m = x.shape[0]
  for i in range(m):
    tmp[i,i] = 1/l[i,i]
    for j in range(i):
      tot = np.dot(tmp[i, j:i], tmp[j, j:i])
      tmp[j,i] = -np.dot(tmp[i, j:i], tmp[j, j:i])/l_ii
      tmp[i,j] = 0
```

Example code

cont.

```
  x_inv = np.zeros((m,m))
  for i in range(m):
    dii = np.dot(tmp[i,i:], tmp[i,i:])
    x_inv[i,i] = dii
    for j in range(i+1, m):
      dij = np.dot(tmp[i, j:], tmp[j, j:])
      x_inv[i,j] = dij
      x_inv[j,i] = dij
  return x_inv
```

Assumptions

ℹ **The expectation function is correct:** the main goal, find a model that predicts the natural phenomena. It can be an iterative process as we update the model after post analysis of our current expectation.

Assumptions

The response is expectation function plus disturbance: this allows us to calculate the probability of the response by means of the probability of the disturbance.

Assumptions

The disturbance is independent of the expectation function: any important variables which are not included in the data are not systematically related to the response.

Assumptions

Each disturbance has a normal distribution: allows us to use least squares. Other distributions are possible as we will see later in this course.

Assumptions

Each disturbance has zero mean: this reduces the number of unknown parameters and implies no systematic bias in the disturbances caused by an unknown influential variable.

Assumptions

ℹ **The disturbance has equal variance:** it simplifies our analysis as the parameters can be estimated independently from σ^2.

Assumptions

The disturbances are distributed independently: this simplifies our join-likelihood function as the product of independent probabilities.

Plotting Residuals

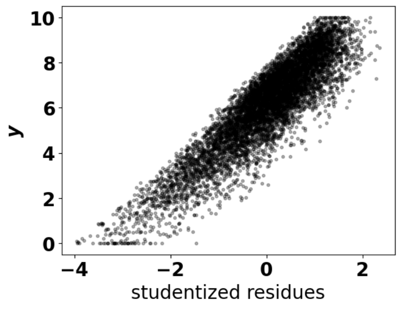
We can calculate the studentized residues as:

$$\frac{y^i - \hat{y}^{(i)}}{s\sqrt{1 - h_{ii}}}$$

where h_{ii} is the i-th diagonal of the matrix $h = x(x^T x)^{-1} x^T = q_1 q_1^T$.

Plotting Residuals

We can plot these values against any quantity of interest such as the predictor variables, time of the experiment, or even $\hat{y}$.



Stabilizing Variance

If there is a relationship in this plot, we can transform the outcomes with a power transformation, such as:

$$y^(\lambda) = \begin{cases} \frac{y^\lambda - 1}{\lambda} & \lambda \neq 0 \\ \ln y & \lambda = 0 \end{cases}$$

Evaluating Regression Models

A straightforward evaluation criteria is to calculate the **Residual Sum of squares** (RSS):

$$RSS = \sum_{i=1}^{N} (y^{(i)} - f(x^{(i)}; \beta))^2$$

Evaluating Regression Models

A better interpretation of the expected error can be achieved with the **Mean Squared Error** (MSE):

$$
\begin{aligned}
MSE &= \mathcal{E}[(y - f(x;\beta))^2] \\
&= \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - f(x^{(i)};\beta))^2
\end{aligned}
$$

Evaluating Regression Models

The MSE can be decomposed into two different terms, called the **variance** and the **bias**:

$$\begin{aligned} MSE &= \mathcal{E}[(y - f(x;\beta))^2))^2] \\ &= \mathrm{Var}[y - f(x;\beta)] + \left(\mathcal{E}[y - f(x;\beta)]^2\right) \\ &= \mathrm{Var}[f(x;\beta)] + \mathrm{Bias}(y, f(x;\beta))^2 \end{aligned}$$

Bias-Variance tradeoff

The variance of the estimator is connected to the sensitivity of the model to the variation in points of the sample.

When the model fits all the points in the data set, the variance will be high. This can indicate that it is also fitting the noise term in our model.

We call this situation **overfitting**.

Bias-Variance tradeoff

When the bias is high, it means that the regression model is returning a high prediction error, thus missing the relationship between predictors and dependent variable.

If this situation happens we say the model is **underfitting**.

It may be necessary to add new predictors.

Coefficient of Determination (R^2)

The **Coefficient of Determination** (R^2) measures the amount of variation in the dependent variable (y) that is captured by the independent variables (x).

Coefficient of Determination (R^2)

Given the residual sum of squares (RSS) and the total sum of squares (TSS):

$$RSS = \sum_{i=1}^{N} (y^{(i)} - f(x^{(i)}; \beta))^2$$

$$TSS = \sum_{i=1}^{N} (y^{(i)} - \mathcal{E}[y]))^2$$

Coefficient of Determination (R^2)

The Coefficient of determination is calcualted as:

$$R^2 = 1 - \frac{RSS}{TSS}$$

If the model perfectly predicts the samples, $R^2 = 1$. If the model always predicts the average value $\mathcal{E}[y]$ (called baseline model), $R^2 = 0$. If it is worse than the baseline, $R^2 < 0$.

Coefficient of Determination (R^2)

In linear regression, whenever we add a new predictor the $R^2 will either be the same or be increased. To compensate for an artificial inflation of the R^2 we can adjust the value as:

$$\begin{aligned}\bar{R}^2 &= 1 - (1 - R^2)\frac{N-1}{N-P-1} \\ &= 1 - (1 - 1 + \frac{RSS}{TSS})\frac{N-1}{N-P-1} \\ &= 1 - \frac{(N-1)RSS}{(N-P-1)TSS}\end{aligned}$$

Regularization

If we are not careful during our experiment design, we may add irrelevant features or correlated features.

For example, our variables **ETA** and **distance** are correlated as they are proportional.

We can also add variables that seem significant but contains random noise, those can help to overfit the data.

Regularization

The problem with correlated features is that it can inflate the coefficients.

Let us take the following model as an example:

$$f(x;\beta) = \beta_1 + \beta_2 x_1 + \beta_3 x_2$$

Regularization

If x_1, x_2 have exactly the same values, the solution making either β_2 or β_3 equal to 0 would generate the same model. Assuming that for $\beta_2 = 0$ the least squares return $\beta_3 = c$.

$$f(x; \beta) = \beta_1 + \beta_2 x_1 + \beta_3 x_2$$

As such, β_2, β_3 can assume any arbitrary value as long as $\beta_2 + \beta_3 = c$. This often leads to a large absolute value for both coefficients.

Regularization

One way to alleviate this problem is to change the least square objective-function adding a penalization factor.

The main regularizations are **Ridge ($l2$)** and **Lasso ($l1$)**.

Ridge ($l2$)

In Ridge regularization we add the following term to the least squares:

$$(y - f(x;\beta))^2 + \lambda \sum_{\mathbf{j=1}}^{\mathbf{P}} \beta_{\mathbf{j}}^{\mathbf{2}}$$

The penalization terms penalizes very high values for the parameters.

Ridge ($l2$)

We have a closed form solution for this regularization as:

$$\hat{\beta} = (x^T x + \lambda I)^{-1} x^T y$$

Lasso ($l1$)

In Lasso we penalize the absolute value of the parameters:

$$(y - f(x;\beta))^2 + \lambda \sum_{\mathbf{j=1}}^{\mathbf{P}} |\beta_{\mathbf{j}}|$$

This penalization does not have a closed-form solution requiring a different optimization algorithm.

It will try to force parameters to 0, serving as a model selection.

Terminology learned today

- **regression analysis:** determining the impact of measurable variables on a phenomena
- **linear regression:** a regression model that tries to fit a linear relationship between independent and dependent variables
- **likelihood function:** a function that returns how likely it is to observe certain parameters values given an observation
- **least squares:** estimation of the linear parameters by minimizing the sum of square residues.
- **statistical test:** answers whether a variable is relevant to the model.
- **null hypothesis:** the hypothesis that a certain parameter is zero.
- **confidence interval:** the range of values of a parameter that rejects the null hypothesis at a certain level.

Terminology learned today

- *Residual Sum of squares** (RSS): sum of the squared residulas of the predictions.
- **Mean Squared Error** (MSE): estimator of the error, can be decomposed in variance and bias.
- **Overfitting:** when the variance is high and the model is fitting noise.
- **Underfitting:** when the bias is high and the model is not fitting the data.
- **Coefficient of Determination (R^2):** measures the variation in the dependent variable that is captures by the dependent variables.
- **Regularization:** a penalization to the least squares that prevent overfitting, alleaviates the effect of correlated variables and remove irrelevant predictors.

Further reading

- Chapter 1 of Bates, Douglas. “Nonlinear regression analysis and its applications.” Wiley Series in Probability and Statistics (1988).
- Chapters 6, 7, 8 of Gelman, Andrew, Jennifer Hill, and Aki Vehtari. Regression and other stories. Cambridge University Press, 2020.

Next lecture

- Nonlinear regression
- Symbolic Regression

Acknowledgments

- Thiago Ferreira Covões