Data-based Statistical Decision Model

Lecture 3 - Model checking and inference

Sungkyu Jung

Model checking or so-called diagnostics

We have laid out what regression analysis is for, why we use statistical models to do it, the assumptions of the simple linear regression model, and estimation and prediction for the simple regression model using both the method of maximum likelihood and the method of least squares.

We will look at the model checking first, then delve into inference.

Model checking is simply to *see* if the assumed model *seems* right.

Recall: Simple Linear Regression Model assumptions

1. The distribution of X is unspecified, possibly even deterministic;
2. $Y|X = \beta_0 + \beta_1 x + \varepsilon$,where ε is a noise variable;
3. ε has mean 0, a constant variance σ^2 (or $\epsilon \sim N(0, \sigma^2)$);
4. ε is uncorrelated with X and uncorrelated across observations.

Residuals are the key in diagnostics

- the residual at the ith data point is the *in-sample prediction error*:

$$\hat{\varepsilon}_i = e_i = Y_i - \hat{m}(X_i),$$

- For the simple linear regression model, it is just

$$e_i = Y_i - (\hat{\beta}_0 + \hat{\beta}_1 X_i).$$

Note:

1. $\sigma^2 = E(\epsilon^2) = MSE(\beta_0, \beta_1)$
2. $\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^n e_i^2 = \text{in-sample } MSE(\hat{\beta}_0, \hat{\beta}_1)$

- The model assumptions are on
    1. the form of $m(x)$ and
    2. the noise variables ϵ_i.
- *If* $m(x) = \beta_0 + \beta_1 x$ *is true*, then the residuals e_i behave like the noises ϵ_i, because

$$e_i = (\beta_0 - \hat{\beta}_0) + (\beta_1 - \hat{\beta}_1)x_i + \epsilon_i,$$

where $(\beta_0 - \hat{\beta}_0)$ and $(\beta_1 - \hat{\beta}_1)$ are hopefully small.

- Thus,
  - if e_i's behave like the model assumptions dictate, then *okay*;
  - if not, then either the assumptions on $m(x)$ or on ϵ_i are violated, and the linear model is (probably) wrong.

More on the residuals

To be more concrete, let's look at what e_i should look like.

$$\begin{aligned} e_i &= \epsilon_i + \frac{1}{n}\sum_{j=1}^{n}\left[1 + (x_i - \bar{x})\frac{x_j - \bar{x}}{s_X^2}\right]\epsilon_j \\ &= \sum_{j=1}^{n}\left[\delta_{ij} + \frac{1}{n} + (x_i - \bar{x})\frac{x_j - \bar{x}}{n s_X^2}\right]\epsilon_j \\ &= \sum_{j=1}^{n} c_{ij}\epsilon_j \end{aligned}$$

Since $E(\epsilon \mid X = x) = 0$, $Var(\epsilon \mid X = x) = \sigma^2$, and they are uncorrelated with each other and with X,

1. $E(e_i \mid X) = 0$
2. $Var(e_i \mid X) = \sigma^2 \sum_{j=1}^{n} c_{ij}^2 = \frac{n-2}{n}\sigma^2 \approx \hat{\sigma}^2$
3. If ϵ_i was Gaussian, then e_i is also Gaussian.

> What about correlations with each other and with X?

To see these, compare ϵ_i and e_i.

- While $E(\sum_i \epsilon_i) = 0$, $\sum_i \epsilon_i \neq 0$. But $\sum_i e_i = 0$.
- While $Cov(\epsilon, X) = 0$, $\sum_i \epsilon_i(X_i - \bar{X}) \neq 0$. But $\sum_i e_i(x_i - \bar{x}) = 0$.
- While $Cov(\epsilon_i, \epsilon_j) = 0$, $Cov(e_i, e_j) \neq 0$. But, $Cov(e_i, e_j) \approx 0$, especially for large n.

These imply that even when the ε's are independent, the residuals are not. However, the dependence is typically very slight and subtle, and it gets weaker as n grows.

Summary on the properties of the Residuals

1. The residuals should have expectation zero, conditional on x, $E[e_i|X = x] = 0$. (The residuals should also have an overall sample mean of exactly zero.)
2. The residuals should show a constant variance, unchanging with x.
3. The residuals can't be completely uncorrelated with each other, but the correlation should be extremely weak, and grow negligible as $n \to \infty$.
4. If the noise is Gaussian, the residuals should also be Gaussian.

- To see if the model assumption is wrong, we want to see if any of the four points above is not true. This is model checking.

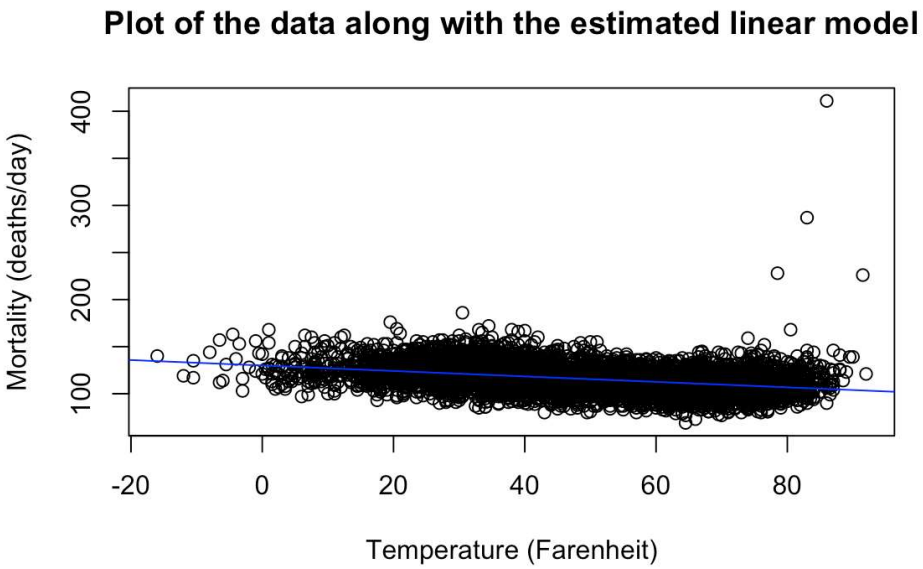
Residual vs predictor (e_i vs x_i) plot

Make a scatter-plot with the residuals on the vertical axis and the predictor variable on the horizontal axis. Because $E[e|X=x]=0$, and $Var[e|X=x]$ is constant, this should, ideally, look like a constant-width blur of points around a straight, flat line at height zero.

- Curved or stepped patterns indicate that $E[e|X=x] \neq 0$, which in turn means that $E[\varepsilon|X=x] \neq 0$, which means that the simple-linear part of the simple linear regression model is wrong.
- Changing the width of the blur indicates that $Var[e|X=x]$ is not constant, which in turn mean that $Var[\epsilon|X=x]$ is not constant.
- Any of the violation is also a sign of ϵ not being independent with X.

As an example, take a look at `chicago` dataset in the package `gamair`.

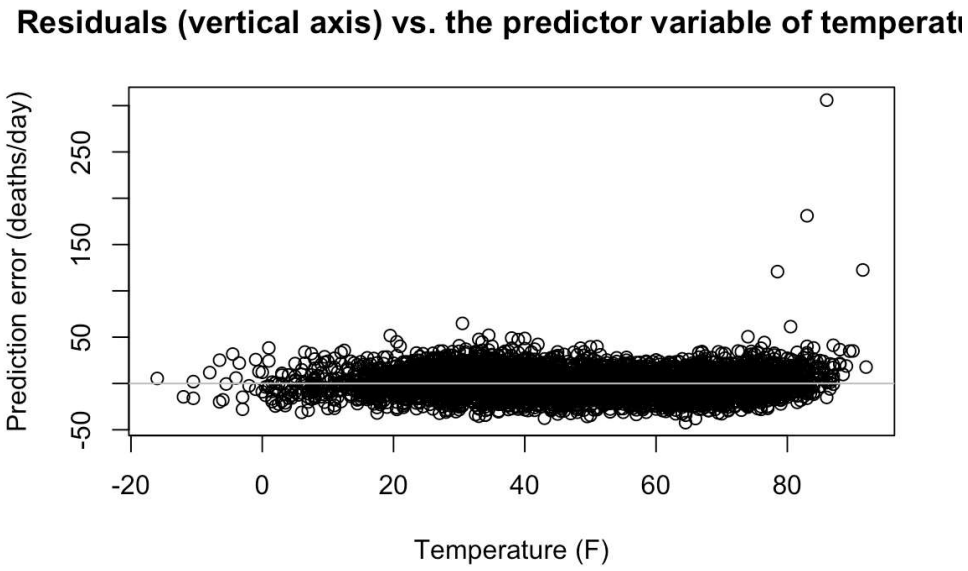
```
 # Load the data set
library(gamair)
data(chicago)
# Plot deaths each day vs. temperature
plot(death ~ tmpd, data=chicago,
     xlab="Temperature (Farenheit)",
     ylab="Mortality (deaths/day)",
     main = "Plot of the data along with the estimated linear model")
# Estimate and store a linear model
death.temp.lm <- lm(death ~ tmpd, data=chicago)
abline(death.temp.lm, col="blue")
```



This is just a plot of data overlaid with the regression line.

Now the residual vs predictor plot:

```
# Always plot residuals vs. predictor variable
plot(chicago$tmpd, residuals(death.temp.lm),
     xlab="Temperature (F)",
     ylab="Prediction error (deaths/day)",
     main = "Residuals (vertical axis) vs. the predictor variable of temperature")
abline(h=0, col="grey")
```



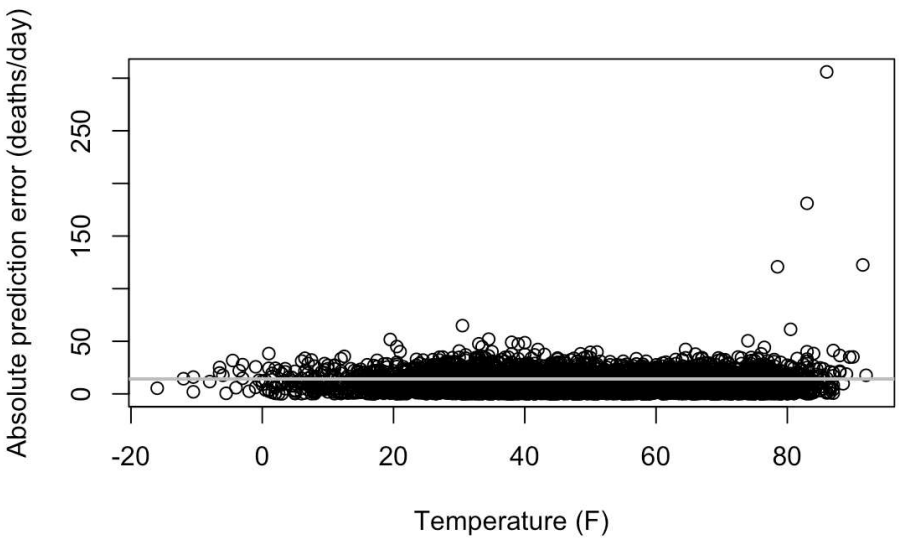
Plot the Magnitude of the Residuals Against the Predictor (e_i^2 vs x_i)

Since $ Var [e|X = x] = E [e^2|X = x]$, we can check whether the variance of the residuals is constant by plotting the squared residuals against the predictor variable. The plot should give a scatter of points around a flat line, whose height should be around the in-sample MSE.

- Regions of the x axis where the residuals are persistently above or below this level are evidence of a problem with the simple linear regression model.
- This could be due to non-constant noise variance, or due to getting the functional form of the regression wrong.

Sometimes, particularly when the model is not doing so well, squaring the residuals leads to a visually uninformative plot. A common fix is to then plot the absolute value of the residuals, with the reference horizontal line being at the square root of the mean squared error.

```
plot(chicago$tmpd, abs(residuals(death.temp.lm)),
     xlab="Temperature (F)",
     ylab="Absolute prediction error (deaths/day)")
abline(h=sqrt(mean(residuals(death.temp.lm)^2)),
       lwd=2, col="grey")
```



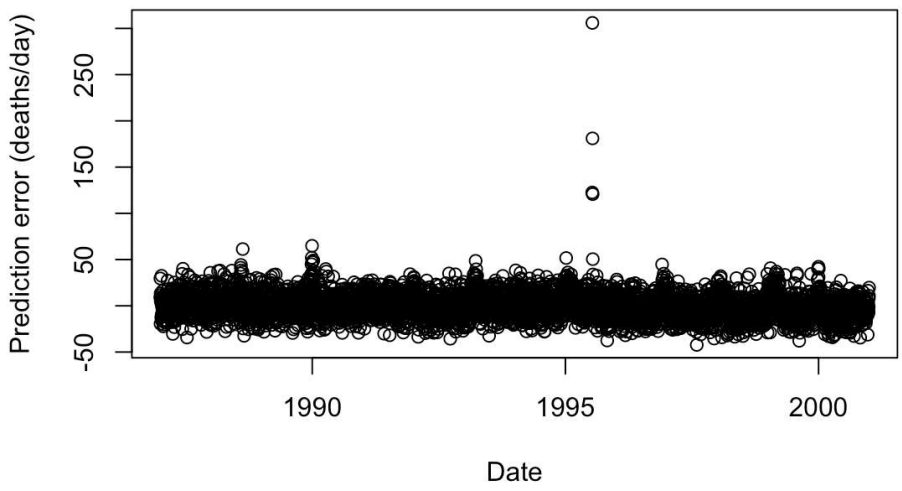
Finding the source of other systematic laws of noise

The residual vs predictor plot should give a general impression of “goodness-of-fit”. However, there may be more sources where the model did not correctly include.

- If you have other potential predictor variables, you should be able to plot the residual against them, and also see a flat line around zero. If not, that’s an indication that the other variable does in fact help predict the response, and so you should probably incorporate that in your model.
- Residuals vs fitted (e_i vs $\hat{m}(x_i)$) is also used oftentimes.
- Lots of the time, our data were collected in a certain order — each data point has some coordinates, in space or in time or both. Under the simple linear regression model, these shouldn’t matter so you should always plot the residuals against the coordinates. Clusters of nearby observations with unusually high or low residuals are a bad sign.

Residuals vs dates:

```
# Always plot residuals vs. coordinate
plot(as.Date(chicago$time,origin="1993-12-31"),
     residuals(death.temp.lm), xlab="Date",
     ylab="Prediction error (deaths/day)")
```



Check whether the residuals are correlated with each other

Residuals vs. Residuals (e_i vs e_{i+1})

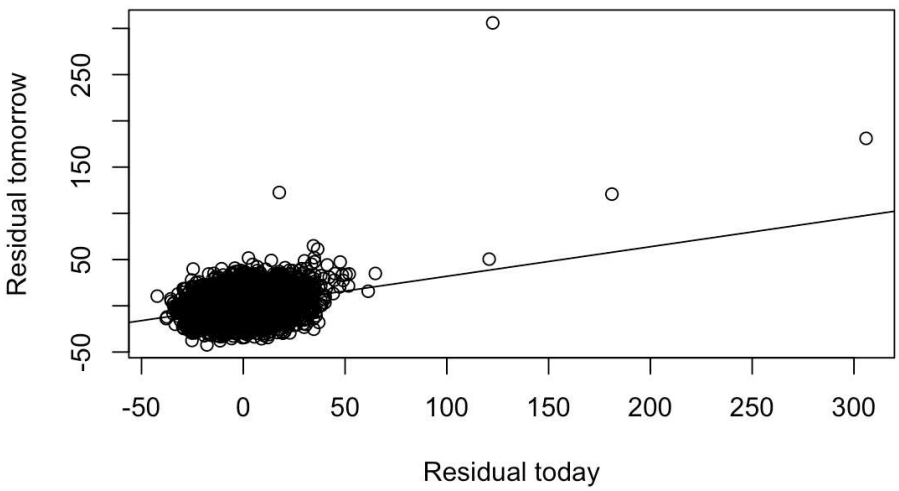
Make a scatter-plot with one point for each observation except the very last; the horizontal coordinate comes from that point's residual, and the vertical coordinate comes from the next point's residual. This is particularly useful when the observations are taken at regular intervals along some axis.

- If the residuals are Gaussian, follow any other bell-ish distribution, it should show a circular blob.
- Falling along a line or curve, or even a tilted ellipse, would be an indication of correlation between successive residuals, which in turn may be a sign that the noise is correlated.

---

residual vs residual plot (lag = 1 day):

```
# Always plot successive residuals against each other
  # head() and tail() here are used to get "every day except the last"
  # and "every day except the first", respectively
  # see help(head)
plot(head(residuals(death.temp.lm),-1),
     tail(residuals(death.temp.lm),-1),
     xlab="Residual today",
     ylab="Residual tomorrow")
abline(lm(tail(residuals(death.temp.lm),-1) ~ head(residuals(death.temp.lm),-1)))
```



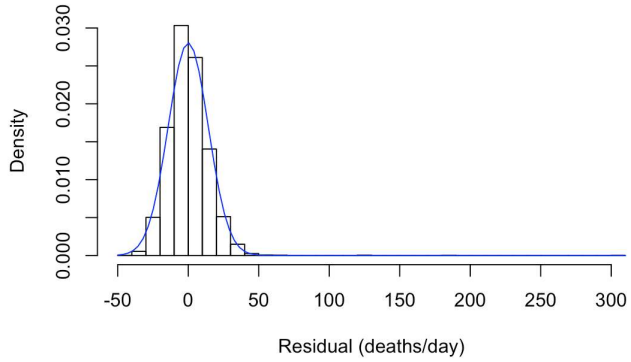
Check whether the Gaussian assumption holds

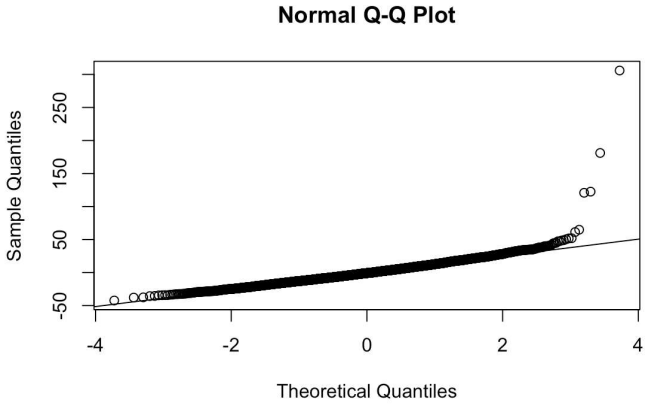
To see if the noises are really Gaussian,

- Approximate the distribution of e ($\approx$ that of ϵ) by the histogram of e_i;
- Use Q-Q plot (plot the residuals against “best-fitted” Gaussian distribution)

```
# Always plot distribution of residuals
hist(residuals(death.temp.lm),breaks=40,
     freq=FALSE,
     xlab="Residual (deaths/day)", main="")
curve(dnorm(x,mean=0,sd=sd(residuals(death.temp.lm))),
      add=TRUE,col="blue")

 # An alternative: plot vs. theoretical Gaussian distribution
qqnorm(residuals(death.temp.lm))
qqline(residuals(death.temp.lm))
```





Generalization errors

- If the model assumptions are correct, it should be able to work about equally well on new data from the same source.

- An important basic check on the model is therefore to divide the data into two parts, estimate the model on one part, the training set, and then examine the predictions and the residuals on the rest of the data, the testing set.
- We can either make the division into training and testing sets by random sampling, or systematically.

Example: Random division

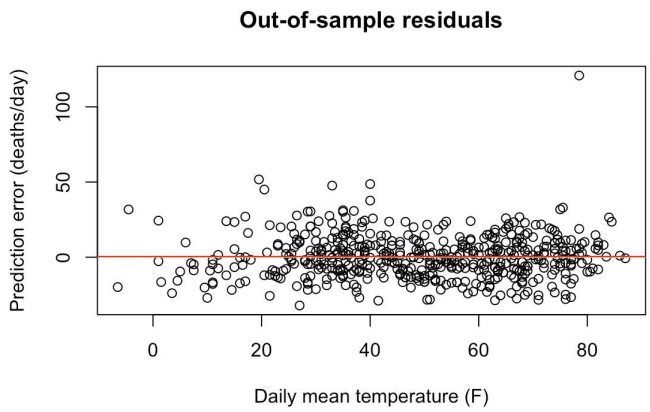
```
# Always look at whether the model can extrapolate to
# new data
# Basic check: randomly divide into two parts, here say 90% of the data # vs. 10%
# Use the "training set" to estimate the model
training.rows <- sample(1:nrow(chicago), size=round(nrow(chicago)*0.9),
                        replace=FALSE)
training.set <- chicago[training.rows,]
# We'll use the "testing set" to see how well it does
testing.set <- chicago[-training.rows,]
# Estimate the model on the training set only
training.lm <- lm(death ~ tmpd, data=training.set)
  # Make predictions on the testing set
  # The model didn't get to see these points while it was being
  # estimated, so this really checks (or tests) whether it can
  # predict
testing.preds <- predict(training.lm, newdata=testing.set)
# Unfortunately residuals() doesn't know about the new data set # so calculate the residuals by hand
testing.residuals <- testing.set$death-testing.preds
```

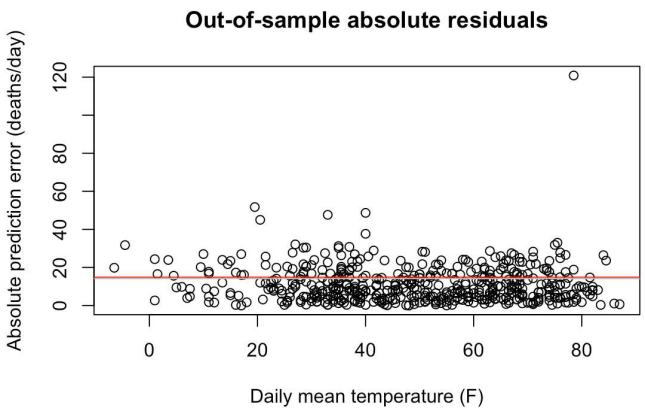
Looking at the out-of-sample residuals

Remember that the data points here were not available to the model during estimation. The grey line marks the average we'd see on the training set (zero), while the red line shows the average on the testing set.

```
# Plot our residuals against the predictor variable
plot(testing.set$tmpd, testing.residuals,
     xlab="Daily mean temperature (F)",
     ylab="Prediction error (deaths/day)",
     main="Out-of-sample residuals")
abline(h=0,col="grey")
abline(h=mean(testing.residuals),col="red")

# Plot absolute residuals vs. predictor variable
plot(testing.set$tmpd, abs(testing.residuals),
     xlab="Daily mean temperature (F)",
     ylab="Absolute prediction error (deaths/day)",
     main="Out-of-sample absolute residuals")
abline(h=sqrt(mean(residuals(training.lm)^2)),col="grey")
abline(h=sqrt(mean(testing.residuals^2)),col="red")
```





Extrapolative Generalization

An alternative to random division, which can be even more useful for model checking, is to systematically make the testing set into a part of the data where the over-all fit of the model seems dubious based on other diagnostics.

Setting up a division of the data into a low-temperature training set and a high-temperature testing set.

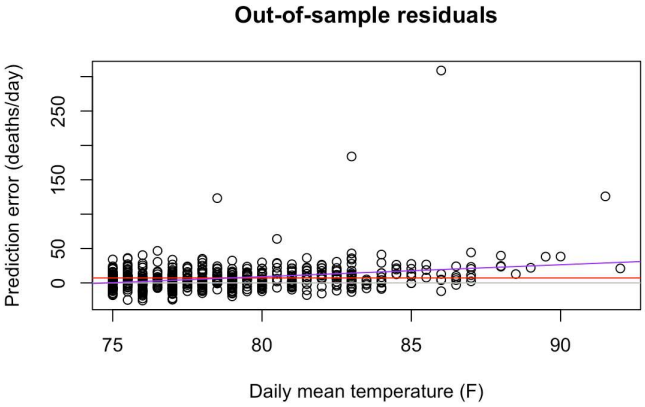
```
# Find the low-temperature days
lowtemp.rows <- which(chicago$tmpd < 75) # About 90% of the data # Divide into low- and high- temperat
ure data sets
lowtemp.set <- chicago[lowtemp.rows,]
hightemp.set <- chicago[-lowtemp.rows,]
# Estimate the model on the colder days only
lowtemp.lm <- lm(death ~ tmpd, data=lowtemp.set)
   # For you: how much do the parameters change, as compared to
   # using all the data?
# Now predict on the high-temperature days
  # Again, these are new data points, but now systematically
  # different (because of their temperature) from the
  # data used to estimate
hightemp.preds <- predict(lowtemp.lm, newdata=hightemp.set) # Calculate our own residuals
hightemp.residuals <- hightemp.set$death-hightemp.preds
```

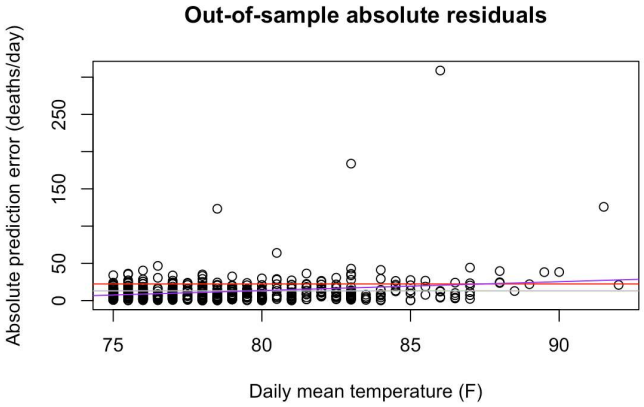
Plot of residuals vs predictors for the testing set.

We have fit the model to low temperatures. We might see whether it can extrapolate to high temperatures, or whether it seems to make systematic errors there.

```
# Plot residuals vs. temperature
plot(hightemp.set$tmpd, hightemp.residuals,
     xlab="Daily mean temperature (F)",
     ylab="Prediction error (deaths/day)",
     main="Out-of-sample residuals")
# Flat line at 0 (ideal, if the model is right)
abline(h=0,col="grey")
# Flat line at the mean of the new residuals
abline(h=mean(hightemp.residuals),col="red")
# Regressing the new residuals on temperature does not look good...
abline(lm(hightemp.residuals ~ hightemp.set$tmpd),col="purple")

# Similar plots for the absolute residuals
plot(hightemp.set$tmpd, abs(hightemp.residuals),
     xlab="Daily mean temperature (F)",
     ylab="Absolute prediction error (deaths/day)",
     main="Out-of-sample absolute residuals")
abline(h=sqrt(mean(residuals(lowtemp.lm)^2)),col="grey")
abline(h=sqrt(mean(hightemp.residuals^2)),col="red")
abline(lm(abs(hightemp.residuals) ~ hightemp.set$tmpd),col="purple")
```





What to do when there is a violation?

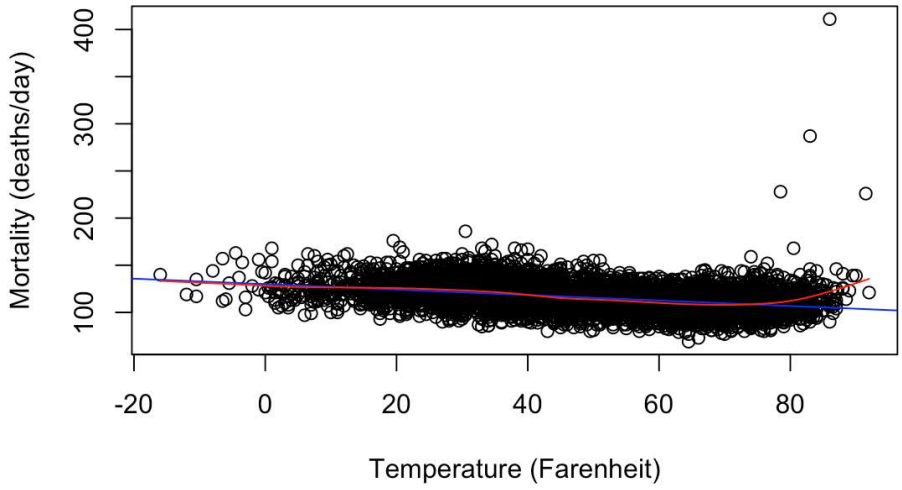
Nonlinear form of X-Y relation

If there's curved or stepped pattern in "e_i vs X_i" plot, the relation between Y and X might not be linear. We may use an alternative model:

1. Linear-after-transformation: e.g. $y = \beta_0 + \beta_1 f(x) + \epsilon$ for some simple $f(\cdot)$.
    - becomes again a simple linear regression model
2. Add more predictors: e.g. $y = \beta_0 + \beta_1 x + \beta_2 z + \epsilon$
    - Multiple linear regression
3. Non-linear but known form: e.g. $y = \beta_0 x^{\beta_1} + \epsilon$.
    - Nonlinear least-square
4. Nonparametrically non-linear form: $y = m(x) + \epsilon$, where $m(x)$ looks nice (or smooth).
    - Local regression
    - Spline smoothing

A smoothing spline example:

```
plot(death ~ tmpd, data=chicago,
     xlab="Temperature (Farenheit)",
     ylab="Mortality (deaths/day)")
death.temp.ss <- smooth.spline(x=chicago$tmpd, y=chicago$death, cv=TRUE)
abline(death.temp.lm, col="blue")
lines(death.temp.ss, col="red")
```



Non-constant noise variance

calls for weighted least squares

Correlated noises

calls for generalized least squares

When Gaussianity fails

Inference either by “large-n” approximation (CLT); or by bootstrap.

Inference

We now do inference on the fitted linear regression model, meaning that

- We add a margin of uncertainty on the point estimators,
- Through specifying the sampling distributions of the estimators,
- And obtain the confidence interval for the unknown true parameters,
- which justfying the hypothesis test procedures,
- And quantify the uncertainty in the prediction of Y.

Standard errors of $\hat{\beta}_1, \hat{\beta}_0$

- The standard error is simply the root of variance:

$$se(\hat{\beta}_0) = \sqrt{\frac{\sigma^2}{n}(1 + \bar{x}^2/s_X^2)}$$

$$se(\hat{\beta}_1) = \sqrt{\frac{\sigma^2}{ns_X^2}}$$

- These are useless (why?) and we might replace σ^2 by $s^2 = \frac{n}{n-2}\hat{\sigma}^2$.

$$\widehat{se}(\hat{\beta}_0) = \sqrt{\frac{\hat{\sigma}^2}{n-2}(1 + \bar{x}^2/s_X^2)}$$

$$\widehat{se}(\hat{\beta}_1) = \sqrt{\frac{\hat{\sigma}^2}{(n-2)s_X^2}}$$

Sampling distributions of $\hat{\beta}_1, \hat{\beta}_0, \hat{\sigma}^2$

Assuming normality, we get the sampling distributions.

- From $\hat{\beta}_1 = \beta_1 + \frac{1}{n}\sum_{i=1}^n \frac{(x_i - \bar{x})}{s_X^2}\epsilon_i$, we get

$$\hat{\beta}_1 \sim N(\beta_1, \frac{\sigma^2}{ns_X^2}).$$

- From $\hat{\beta}_0 = \beta_0 + \frac{1}{n}\sum_{i=1}^n (1 + \frac{(x_i - \bar{x})}{s_X^2})\epsilon_i$, we get

$$\hat{\beta}_0 \sim N(\beta_0, \frac{\sigma^2}{n}(1 + \bar{x}^2/s_X^2)).$$

- From $\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^n e_i^2 = \frac{1}{n}\sum_{i=1}^n (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2$, we get

$$n\frac{\hat{\sigma}^2}{\sigma^2} \sim \chi^2_{n-2}.$$

Let's focus on $\hat{\beta}_1$

- We have $\hat{\beta}_1 \sim N(\beta_1, se(\hat{\beta}_1))$.
- The sampling distribution of the difference seems more useful:

$$\hat{\beta}_1 - \beta_1 \sim N(0, se(\hat{\beta}_1)),$$

  for any β_1!

- Since we do not know σ^2 (nor $se(\hat{\beta}_1)$), replace $se(\hat{\beta}_1)$ by $\widehat{se}(\hat{\beta}_1)$. Then

$$\frac{\hat{\beta}_1 - \beta_1}{\widehat{se}(\hat{\beta}_1)} \sim t_{n-2}.$$

(For large n, $t_{n-2} \approx N(0,1)$.)

This formula contains only one unknown parameter, β_1, and is used in finding the exact confidence interval for β_1 and evaluating the power of tests about β_1. If $n \gg 30$ (large), then we can say:

- The difference is at most $1.96\widehat{se}(\hat{\beta}_1)$ with 95% chance.

- Equivalently, the estimate $\hat{\beta}_1$ should be in the interval $\beta_1 \pm 1.96\widehat{se}(\beta_1)$ with 95% of time.

- Equivalently, the unknown β_1 is in the interval $\hat{\beta}_1 \pm 1.96\widehat{se}(\hat{\beta}_1)$ with 95% chance.

A test for β_1

To be more general, let $k = k(n, \alpha)$ be such that

$$\int_{-k}^{k} f_{n-2}(t)dt = 1 - \alpha,$$

where f_{n-2} is the density function of t_{n-2} distribution.

Now the second point above generalizes to

- The estimate $\hat{\beta}_1$ should be in the interval $\beta_1 \pm k(n, \alpha)\widehat{se}(\hat{\beta}_1)$ with $1 - \alpha$ of time.

This suggests a test procedure: Suppose we test

$$H_0 : \beta_1 = \beta_1^* \quad vs \quad H_1 : \beta_1 \neq \beta_1^*.$$

Reject H_0 if $\hat{\beta}_1$ is not in the interval $\beta_1 \pm k(n, \alpha)\widehat{se}(\hat{\beta}_1)$.

(This is the “t-test” you may have seen before.)

Confidence interval

- A 95% confidence interval $C = (l, u)$ for a parameter β_1 is a random interval (a function of data) that traps β_1 95% of the time.
- The third point above generalizes to

- the unknown β_1 is in the interval $C = \hat{\beta}_1 \pm k(n, \alpha)\widehat{se}(\hat{\beta}_1)$ with with $1 - \alpha$ of time.

Width of the confidence interval

Notice that the width of the confidence interval is $2k(n,\alpha)\widehat{se}(\hat{\beta}_1)$.

1. As α shrinks, the interval widens. (High confidence comes at the price of big margins of error.)
2. As n grows, the interval shrinks. (Large samples mean precise estimates.)
3. As σ^2 increases, the interval widens. (The more noise there is around the regression line, the less precisely we can measure the line.)
4. As s_X^2 grows, the interval shrinks. (Widely-spread measurements give us a precise estimate of the slope.)

Predictive inference

Predict $Y \mid X = x$

1. as a point
2. as a distribution
3. as an interval

Confidence intervals for conditional means

Estimate the conditional mean

$$m(x) \equiv EY|X = x = \beta_0 + \beta_1 x$$

by

$$\hat{m}(x) = \hat{\beta}_0 + \hat{\beta}_1 x.$$

We've seen that

$$\hat{m}(x) = \beta_0 + \beta_1 x + \frac{1}{n}\sum_{i=1}^{n}\left(1 + (x - \overline{x})\frac{x_i - \overline{x}}{s_X^2}\right)\epsilon_i$$

so that

$$E\hat{m}(x) = \beta_0 + \beta_1 x = m(x)$$

and

$$Var\hat{m}(x) = \frac{\sigma^2}{n}\left(1 + \frac{(x - \overline{x})^2}{s_X^2}\right).$$

Under the Gaussian noise assumption, $\hat{m}(x)$ is Gaussian (why?),

$$\hat{m}(x) \sim N\left(m(x), \frac{\sigma^2}{n}\left(1 + \frac{(x - \overline{x})^2}{s_X^2}\right)\right)$$

Exact confidence intervals

At this point, getting confidence intervals for $m(x)$ works just like getting confidence intervals for β_0 or β_1: we use as our standard error

$$\widehat{se}(\hat{m}(x)) = \frac{\hat{\sigma}}{\sqrt{n-2}}\sqrt{1+\frac{(x-\overline{x})^2}{s_X^2}}$$

and then find

$$\frac{\hat{m}(x)-m(x)}{\widehat{se}(\hat{m}(x))} \sim t_{n-2}$$

by entirely parallel arguments. $1-\alpha$ confidence intervals follow as before as well.

Interpreting the confidence interval

This confidence interval has the same interpretation as one for the parameters: either

1. The true value of $m(x)$, i.e., the true value of $EY|X=x$, is in the interval, or
2. Something very unlikely happened when we got our data.

This is all well and good, but it does not tell us about how often future values of Y will be in this interval; it tells us about how often we capture the conditional average.

Prediction Interval

A $1-\alpha$ prediction interval for $Y|X=x$ is a an interval $[l,u]$ where

$$P(l \leq Y \leq u|X=x) = 1-\alpha$$

Since $Y|X=x \sim N(m(x),\sigma^2)$, it would be a simple matter to find these limits if we knew the parameters: the lower limit would be $m(x)+z_{\alpha/2}\sigma$, and the upper limit $m(x)+z_{1-\alpha/2}\sigma$. Unfortunately, we don't know the parameters.

However, we do know how the parameters are related to our estimates, so let's try to use that:

$$\begin{aligned}
Y|X=x &\sim N(m(x),\sigma^2) \\
&= m(x) + N(0,\sigma^2) \\
&= \hat{m}(x) + N\left(0, \frac{\sigma^2}{n}\left(1+\frac{(x-\overline{x})^2}{s_X^2}\right)\right) + N(0,\sigma^2) \\
&= \hat{m}(x) + N\left(0, \sigma^2\left(1+\frac{1}{n}+\frac{(x-\overline{x})^2}{ns_X^2}\right)\right)
\end{aligned}$$

Let's call the variance $\sigma^2_{pred}(x)$.

So, we have a random variable with a Gaussian distribution centered at $\hat{m}(x)$ and with a variance $\sigma^2_{pred}(x)$ proportional to σ^2. We can estimate that variance as

$$s^2_{pred}(x) = \hat{\sigma}^2\frac{n}{n-2}\left(1+\frac{1}{n}+\frac{(x-\overline{x})^2}{ns_X^2}\right)$$

Going through the now-familiar argument once again,

$$\frac{Y - \hat{m}(x)}{s_{pred}(x)} \mid X = x \sim t_{n-2}$$

and we can use this to give prediction intervals.

Interpretation of the prediction interval

The interpretation of the prediction interval here is a bit tricky.

What we want for a prediction interval is that

$$P(l \leq Y \leq u \mid X = x) = 1 - \alpha$$

Now our limits l and u involve the estimated parameters. To be explicit,

$$P(Y \in \hat{m}(x) \pm k(n, \alpha)\widehat{se}(\hat{m}(x)) \mid X = x) = 1 - \alpha$$

It tells you about how often future values of Y will be in this interval when $X = x$;

Numerical computation in R for inference

When we estimate a model with `lm`, R makes it easy for us to extract the confidence intervals of the coefficients:

```
confint(object, level=0.95)
```

Here object is the name of the fitted model object, and level is the confidence level. For instance,

```
library(gamair); data(chicago)
death.temp.lm <- lm(death ~ tmpd, data=chicago)
confint(death.temp.lm) # What happened to the second argument?
```

```
##                  2.5 %     97.5 %
## (Intercept) 128.8783687 131.035734
## tmpd         -0.3096816  -0.269607
```

```
confint(death.temp.lm, level=0.90)
```

```
##                   5 %        95 %
## (Intercept) 129.0518426 130.8622598
## tmpd         -0.3064592  -0.2728294
```

The `summary` function will also print out a lot of information about the model:

```
summary(death.temp.lm)
```

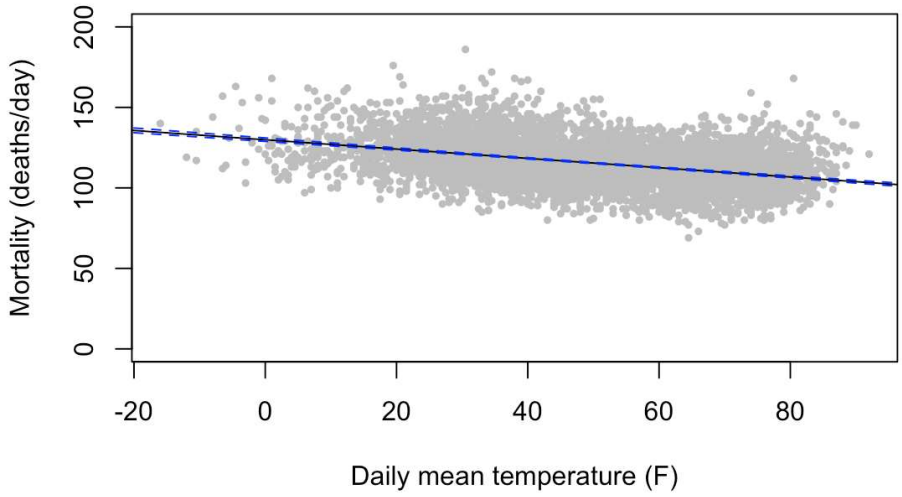
```
##
## Call:
## lm(formula = death ~ tmpd, data = chicago)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.275  -9.018  -0.754   8.187 305.952
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 129.95705    0.55023  236.19   <2e-16 ***
## tmpd         -0.28964    0.01022  -28.34   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.22 on 5112 degrees of freedom
## Multiple R-squared:  0.1358, Adjusted R-squared:  0.1356
## F-statistic: 803.1 on 1 and 5112 DF,  p-value: < 2.2e-16
```

For linear models, all of the calculations needed to find confidence intervals for `m` or prediction intervals for Y are automated into the `predict` function.

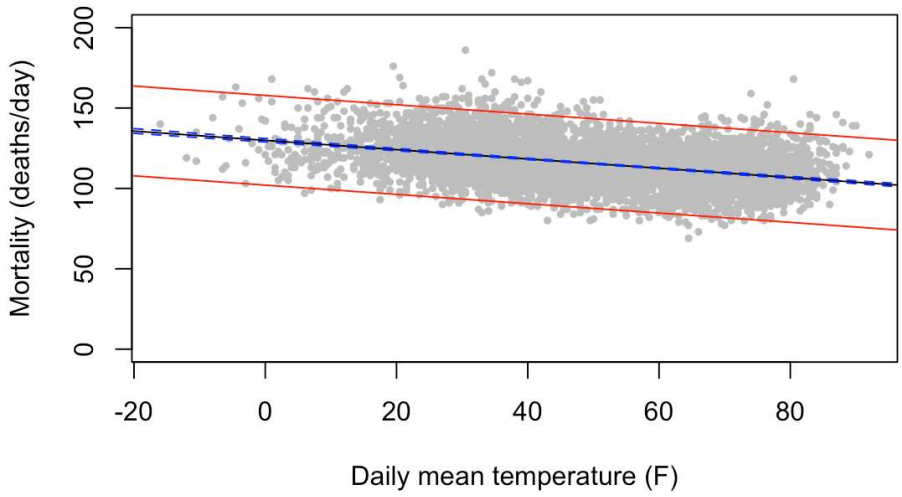
```
predict(object, newdata, interval=c("none", "confidence", "prediction"), level=0.95)
```

Data from the Chicago death example (grey dots), together with the regression line (solid black) and the 95% confidence limits on the conditional mean (dashed blue curves). I have restricted the vertical range to help show the confidence limits, though this means some high-mortality days are off-screen.

```
plot(death ~ tmpd, data=chicago, pch=19, cex=0.5, col="grey", ylim=c(0,200),
     xlab="Daily mean temperature (F)", ylab="Mortality (deaths/day)")
abline(death.temp.lm)
temp.seq <- seq(from=-20, to=100, length.out=100)
death.temp.CIs <- predict(death.temp.lm, newdata=data.frame(tmpd=temp.seq),
                          interval="confidence")
lines(temp.seq, death.temp.CIs[,"lwr"], lty="dashed", col="blue")
lines(temp.seq, death.temp.CIs[,"upr"], lty="dashed", col="blue")
```



Adding 95% prediction intervals (red) to the previous plot.




```
death.temp.Pls <- predict(death.temp.lm, newdata=data.frame(tmpd=temp.seq),
                          interval="prediction")
lines(temp.seq, death.temp.Pls[,"lwr"], col="red")
lines(temp.seq, death.temp.Pls[,"upr"], col="red")
```

F-test and R^2

We have seen in the `summary(object)`, "F-statistic" and "R-squared". Let's briefly see what they are.

F-test

We saw that we could test the null hypothesis that $\beta_1 = 0$ using the statistic $(\hat{\beta}_1 - 0)/\widehat{se}$. (Although confidence intervals are generally more important than testing.) The F-test is another approach of testing "no relation between x and y".

The idea is comparing two models:

	Null model	**Linear model**
Model	$Y = \beta_0 + \epsilon$	$Y = \beta_0 + \beta_1 X + \epsilon.$
Residuals	$y_i - \bar{y}$	$e_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)$
Sum of squares	$\sum_{i=1}^n (y_i - \bar{y})^2$	$\sum_{i=1}^n e_i^2$

- The sum of squares for the null model is often denoted by SS_{total}.
- The sum of squares for the linear model is the S.S. of (usual) residuals, denoted by RSS.
- RSS is always smaller than SS_{total} (why?) and we define *the sum of squares due to regression* by $SS_{reg} = SS_{total} - RSS$.

This gives the so-called ANOVA (Analysis of Variance) table.

Source	df	SS	MS	F	p-value
Regression	1	SS_{reg}	$\text{MS}_{\text{reg}} = \frac{\text{SS}_{\text{reg}}}{1}$	$F = \frac{\text{MS}_{\text{reg}}}{\text{MS}_{\text{res}}}$	
Residual	n-2	RSS	$\widehat{\sigma}^2 = \frac{\text{RSS}}{n-2}$		
Total	n-1	SS_{total}			

- The F-value would be small if SS_{reg} is small (i.e. the regression does nothing).
- The F-value would be large if SS_{reg} is large (i.e. $\beta_1 \neq 0$).

```
anova(death.temp.lm)
```

```
## Analysis of Variance Table
##
## Response: death
##             Df  Sum Sq Mean Sq F value    Pr(>F)
## tmpd         1  162473  162473  803.07 < 2.2e-16 ***
## Residuals 5112 1034236     202
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

R^2

The R-squared is “the fraction of variability explained by the regression”, i.e.

$$R^2 = \frac{SS_{reg}}{SS_{total}}.$$

It is also the correlation coefficient squared (thus r-squared): $R^2 = r^2$, where

$$r = \frac{\widehat{Cov}(X, Y)}{s_X s_Y}.$$

To better understand R^2, we look at its population counterpart:

$$\begin{aligned} R^2 &= \frac{Var(m(X))}{Var(Y)} \\ &= \frac{Var(\beta_0 + \beta_1 X)}{Var(\beta_0 + \beta_1 X + \epsilon)} \\ &= \frac{\beta_1^2 Var(X)}{\beta_1^2 Var(X) + \sigma^2)} \end{aligned}$$

- R^2 is 0 if $\beta_1 = 0$ (or $\sigma^2 \to \infty$).

- R^2 is 1 if $\sigma^2 = 0$ (or $Var(X)$ much bigger than σ^2).

What F-test and R^2 really tell us

1. F-test only works when the Gaussian assumption is true.
2. F-test tests whether our linear model is *better* than the null model.
3. F-test does not tell whether the linear model (or the null model) is a good fit.
4. R^2 does not measure the goodness-of-fit.
    - R^2 can be low even if the model is correct.
    - R^2 can be high even if the model is wrong
5. R^2 says nothing about prediction error