```
[1] "name"       "birthmonth" "birthyear"  "length"     "width"
[6] "sex"        "biggerfoot" "domhand"
```

```
> with(data=feet,levels(sex))
```

```
[1] "B" "G"
```

```
> mod = lm(width ~ length + sex, data=feet)
```

The object created by `lm()` contains a lot of information about the model and the data used for fitting the model. For the purposes of using the model to make a prediction, you want a function that takes the values for the explanatory variables as inputs and returns the corresponding model value as the output. To turn the model into a function, use `makeFun()`. You can call the resulting function anything you want. Here, it's called f:

```
> f = makeFun(mod)
```

Now you can use `f()` to convert inputs into outputs. Take care to use the right coding for categorical variables.

```
> f( length=25, sex="G" )
```

```
   1
8.93
```

In order to generate a confidence interval, the prediction function `f()` needs to be told what type of interval is wanted. There are two types of prediction confidence intervals:

Interval on the model value which reflects the sampling distributions of the coefficients themselves. To calculate this, use the `interval="confidence"` named argument:

```
> f( length=25, sex="G", interval="confidence")
```

```
   fit  lwr  upr
1 8.93 8.74 9.13
```

The components named `lwr` and `upr` are the lower and upper limits of the confidence interval, respectively.

Interval on the prediction which includes the variation due to the uncertainty in the coefficients as well as the size of a typical residual. To find this interval, use the `interval="prediction"` named argument:

```
> f( length=25, sex="G", interval="prediction" )
```

```
   fit  lwr  upr
1 8.93 8.13 9.74
```

The prediction interval is larger than the model-value confidence interval because the residual always gives additional uncertainty around the model value.