Tutorial 09: Prediction assessment with proper scores (solutions)

3D printer

In this tutorial, you will use the data file `"filament1.rda"`, which contains information about one 3D-printed object per row. The columns of this dataset are

- `Index`: an observation index
- `Date`: printing dates
- `Material`: the printing material, identified by its colour
- `CAD_Weight`: the object weight (in gram) that the CAD software calculated
- `Actual_Weight`: the actual weight of the object (in gram) after printing

Estimation and prediction

Load the `filament1.rda` data. The CAD weight for observations i is denoted by $x_{i,1}$, the material is $x_{i,2}$ and the corresponding actual weight is y_i. Consider two linear models, named A and B:

- Model A: $y_i = \theta_0 + \theta_1 x_{i,1} + \epsilon_i$
- Model B: $y_i = \theta_0 + \theta_1 x_{i,1} + \theta_2 x_{i,2} + \epsilon_i$

Create a function that estimates the linear models A and B using the `lm` built in `R` function and then calculates predictions using the `lm.predict` function in `R`. Your function should return a `data.frame` with variables mean, sd, lwr, and upr, summarizing the prediction distribution for each row of the new data.

Solution:

```
lm_prediction <- function(data, model, newdata) {
  if (model == "A") {
    fit0 <- lm(Actual_Weight ~ 1 + CAD_Weight, data = data)

  } else {
     fit0 <- lm(Actual_Weight ~ 1 + CAD_Weight + Material, data = data)
  }

  pred0 <- predict(fit0, newdata = newdata, se.fit = TRUE,
                   interval = "prediction", level = 0.95)

  mean =  pred0$fit[, "fit"]
  sd <- sqrt(pred0$se.fit^2 + summary(fit0)$sigma^2)

    q <- qt(1 - 0.05/2, df = Inf)
    lwr <- mean - q * sd
    upr <- mean + q * sd

  data.frame(mean = mean, sd = sd,
             lwr = lwr, upr = upr)
}
```

Next, use your function to compute probabilistic predictions of `Actual_Weight` using the two estimated models and the `filament1` as new data. Use a level of significance of 5% for computing the prediction intervals. Note that in this exercise, the data for estimating and predicting the models are the same.

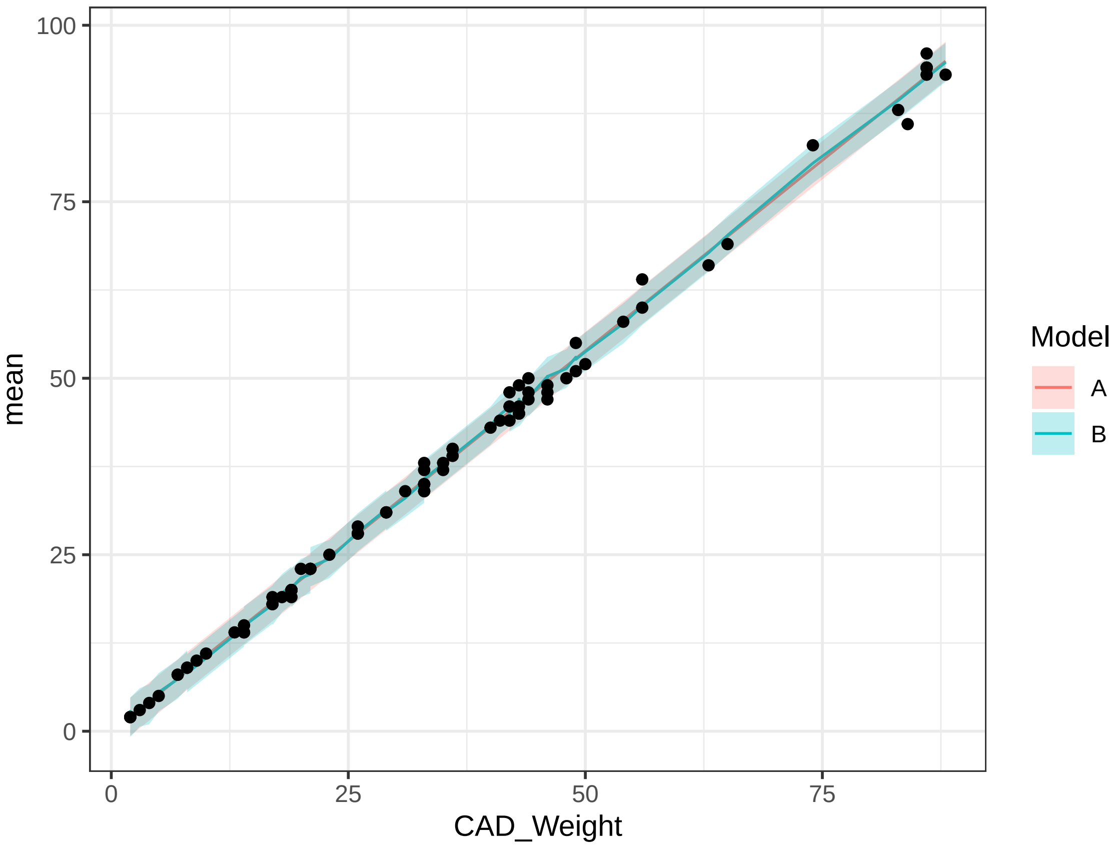
Solution:

```
pred_A <- lm_prediction(data = filament1, model = "A", newdata = filament1)
pred_B <- lm_prediction(data = filament1, model = "B", newdata = filament1)
```

Inspect the predictions by drawing figures, e.g. with `geom_ribbon(aes(CAD_Weight, ymin = lwr, ymax = upr), alpha = 0.25)` (the `alpha` here is for transparency), together with the observed data. It can be useful to join the predictions and data into a new `data.frame` object to get access to both the prediction information and data when plotting.

Solution:

```
ggplot(rbind(cbind(pred_A, filament1, Model = "A"),
             cbind(pred_B, filament1, Model = "B")),
       mapping = aes(CAD_Weight)) +
  geom_line(aes(y = mean, col = Model)) +
  geom_ribbon(aes(ymin = lwr, ymax = upr, fill = Model), alpha = 0.25) +
  geom_point(aes(y = Actual_Weight), data = filament1)
```



Here, the `geom_point` call gets its own `data` input to ensure each data point is only plotted once.

Prediction Scores

Compute the squared error and Dawid-Sebastiani scores for the predictions. It's useful to joint the prediction information and data set with `cbind`, so that e.g. `mutate()` can have access to all the needed information.

Solution:

```
score_A <- cbind(pred_A, filament1) %>%
  mutate(
```

```
    se = (Actual_Weight - mean)^2,
    ds = (Actual_Weight - mean)^2/sd^2 + 2 * log(sd)
  )
score_B <- cbind(pred_B, filament1) %>%
  mutate(
    se = mean((Actual_Weight - mean)^2),
    ds = (Actual_Weight - mean)^2/sd^2 + 2 * log(sd)
  )
```

See the next section for a more compact alternative (first combining the prediction information from both models, and then computing all the scores in one go).

As a basic summary of the results, compute the average score $\overline{S}(\{F_i, y_i\})$ for each model and type of score by following these steps: 1. Join the A-scores with a 'model' variable with `cbind`. 2. Do the same for B, and then join the two with `rbind`. 3. Use `group_by()` and `summarise()` to collect the average scores for each model and each type of score. 4. Display the result with `knitr::kable()`.

Solution:

```
score_AB <-
  rbind(cbind(score_A, model = "A"),
        cbind(score_B, model = "B")) %>%
  group_by(model) %>%
          summarise(se = mean(se),
                    ds = mean(ds))
knitr::kable(score_AB)
```

model	se	ds
A	1.783379	1.571325
B	1.670444	1.517489

Do the scores indicate that one of the models is better or worse than the other? Do the three score types agree with each other?

Solution:

The squared error score doesn't really care about the difference between the two models, since it doesn't directly involve the variance model (the parameter estimates for the mean are different, but not by much). The DS score indicate that model B is better than model A.

Data splitting

Now, use the `sample()` function to generate a random selection of the rows of the data set, and split it into two parts, with ca 50% to be used for parameter estimation, and 50% to be used for prediction assessment. Redo the previous analysis for the problem using this division of the data.

Solution:

```
idx_est <- sample(filament1$Index,
                  size = round(nrow(filament1) * 0.5),
                  replace = FALSE)
filament1_est <- filament1 %>% filter(Index %in% idx_est)
filament1_pred <- filament1 %>% filter(!(Index %in% idx_est))
```

Solution:

```r
pred_A <- lm_prediction(data = filament1, model = "A", newdata = filament1_pred)
pred_B <- lm_prediction(data = filament1, model = "B", newdata = filament1_pred)
scores <-
  rbind(cbind(pred_A, filament1_pred, model = "A"),
        cbind(pred_B, filament1_pred, model = "B")) %>%
  mutate(
    se = (Actual_Weight - mean)^2,
    ds = (Actual_Weight - mean)^2/sd^2 + 2 * log(sd)
  )

score_summary <- scores %>%
  group_by(model) %>%
  summarise(se = mean(se),
            ds = mean(ds))
knitr::kable(score_summary)
```

model	se	ds
A	2.404589	1.902478
B	2.254300	1.816779

Do the score results agree with the previous results?

Solution:

The results will have some more random variability due to the smaller size of the estimation and prediction sets, but will likely agree with the previous results.