Assignment 3

Natali Tckvitishvili

2024-06-20

General Information

- **Points**: Assignment 3 comprises of 6 tasks, 2 points each (12 in total). 2 points are obtained for complete and correct answers. 1 point is obtained for a proper approach or if only part of the task is solved.
- **Submission**: Hand in the assignment as a `Markdown` report (RMarkdown or Quarto) rendered as PDF. The PDF report should show the result(s), the code that produced the result(s), and possibly additional text or comment. Also indicate your name. The report should be uploaded on Moodle until Wednesday, June 21, 9:45 am.
- **Working in teams**: Everyone needs to hand in a report on Moodle. However, the report can be handed in as a team work (max. 2 people). When working in teams, state at the beginning of the document, who you worked with. It Ideally, teams use GitHub and add a link to the GitHub repository to which both contributed.
- **Code**: To automate code wrapping (such that long code lines are not cut off), install the formatR package and add the following code chunk at the beginning of the document:

```
knitr::opts_chunk$set(tidy = TRUE, tidy.opts=list(width.cutoff=50))
```

Load the data set `shaq` to solve the tasks below. If the `Markdown` document and the data set are stored in different folders (e.g., "BayesIntro/assignments/assignment_3.md" and "BayesIntro/data/shaq.csv" you can use the package `here` to load the data.

```
library(dplyr)
library(rethinking)
library(tidyr)
library(ggplot2)
```

```
shaq <- read.csv("shaq.csv")
head(shaq)
```

```
##   Season SeasGm CarrGm  Date     Age  Tm Home Opp Win teamdiff GS Minutes FG
## 1      1      1      1 33914 20.6708 ORL    1 MIA   1       10  1      32  4
## 2      1      2      2 33915 20.6735 ORL    0 WSB   1        5  1      40  8
## 3      1      3      3 33918 20.6817 ORL    1 CHH   0       -4  1      34 15
## 4      1      4      4 33920 20.6872 ORL    1 WSB   1       27  1      36 12
## 5      1      5      5 33922 20.6927 ORL    0 NJN   0      -11  1      35  9
## 6      1      6      6 33926 20.7036 ORL    0 PHI   1       10  1      34 12
##   FGA   FG. X3P X3PA X3P. FT FTA   FT. ORB DRB TRB AST STL BLK TOV PF PTS GmSc
## 1   8 0.500   0    0   NA  4   7 0.571   5  13  18   2   1   3   8  6  12  8.3
## 2  16 0.500   0    0   NA  6  11 0.545   5  10  15   1   0   4   4  5  22 16.0
## 3  25 0.600   0    0   NA  5   8 0.625   4   9  13   1   1   3   4  4  35 26.0
## 4  19 0.632   0    0   NA  7  12 0.583   9  12  21   1   0   4   6  4  31 26.3
## 5  16 0.563   0    0   NA 11  16 0.688   5  10  15   1   1   3   2  4  29 26.1
## 6  19 0.632   0    0   NA  5  11 0.455   7  12  19   1   1   3   3  5  29 25.4
##   Pls.Mns
## 1      NA
## 2      NA
## 3      NA
## 4      NA
## 5      NA
## 6      NA
```

Task Set 1

For Tasks 1.1 and 1.2, create a training data set `shaq_training` that contains all the data from the `Season` 1 to 5.

```
shaq_training <- shaq %>%
  filter(Season <= 5)
```

Task 1.1

Use the training data and estimate a simple regression model where you predict points (PTS) from field goal attempts (FGA). Specify the regression model such that the intercept represents the expected number of points, given an average number of FGA. Provide a table that summarizes the posterior distribution.

```
# mean-centering
FGA_mean <- round(mean(shaq_training$FGA), 0)
```

```
simple_model <- quap(
  alist (
    PTS ~ dnorm(mu, sd),
    mu <- a + b_1 * (FGA - FGA_mean),
    a ~ dnorm(25, 8),
    b_1 ~ dunif(0, 3), # score between 0 and 3
    sd ~ dunif(0, 8)
  ), data = shaq_training
)
precis(simple_model)
```

```
##          mean         sd      5.5%     94.5%
## a   27.241922 0.26761425 26.814223 27.669621
## b_1  1.173308 0.05395662  1.087075  1.259541
## sd   4.977555 0.18921831  4.675148  5.279962
```

Task 1.2

Estimate a multiple regression model, where you add free throw attempts (FTA) as a second predictor. Again, the intercept should represents the expected number of points, given an average number of FGA and FTA. Provide a table that summarizes the posterior distribution.

```
FTA_mean <- round(mean(shaq_training$FTA), 0)
multi_model <- quap(
  alist (
    PTS ~ dnorm(mu, sd),
    mu <- a + b_1 * (FGA - FGA_mean) + b_2 * (FTA - FTA_mean),
    a ~ dnorm(25, 8),
    b_1 ~ dunif(0, 3),
    b_2 ~ dunif(0, 1),
    sd ~ dunif(0, 8)
  ), data = shaq_training
)
precis(multi_model)
```

```
##           mean         sd       5.5%      94.5%
## a   27.3001832 0.23337047 26.9272122 27.6731543
## b_1  1.0495830 0.04849822  0.9720734  1.1270925
## b_2  0.6114536 0.05846931  0.5180084  0.7048989
## sd   4.3388349 0.16493776  4.0752326  4.6024373
```

Task Set 2

For Tasks 2.1 and 2.2, create a training data set `shaq_test` that contains all the data from the `Season` 6 to 10.

```
shaq_test <- shaq %>%
  filter(Season >= 6 & Season <= 10)
```

Task 2.1

Use posterior samples from the simple regression model that you estimated in Task 1.1 and the FGA data from the test set to predict new points. Create a plot that shows the predicted point distribution along the actual point distribution from Season `Season` 6 to 10.

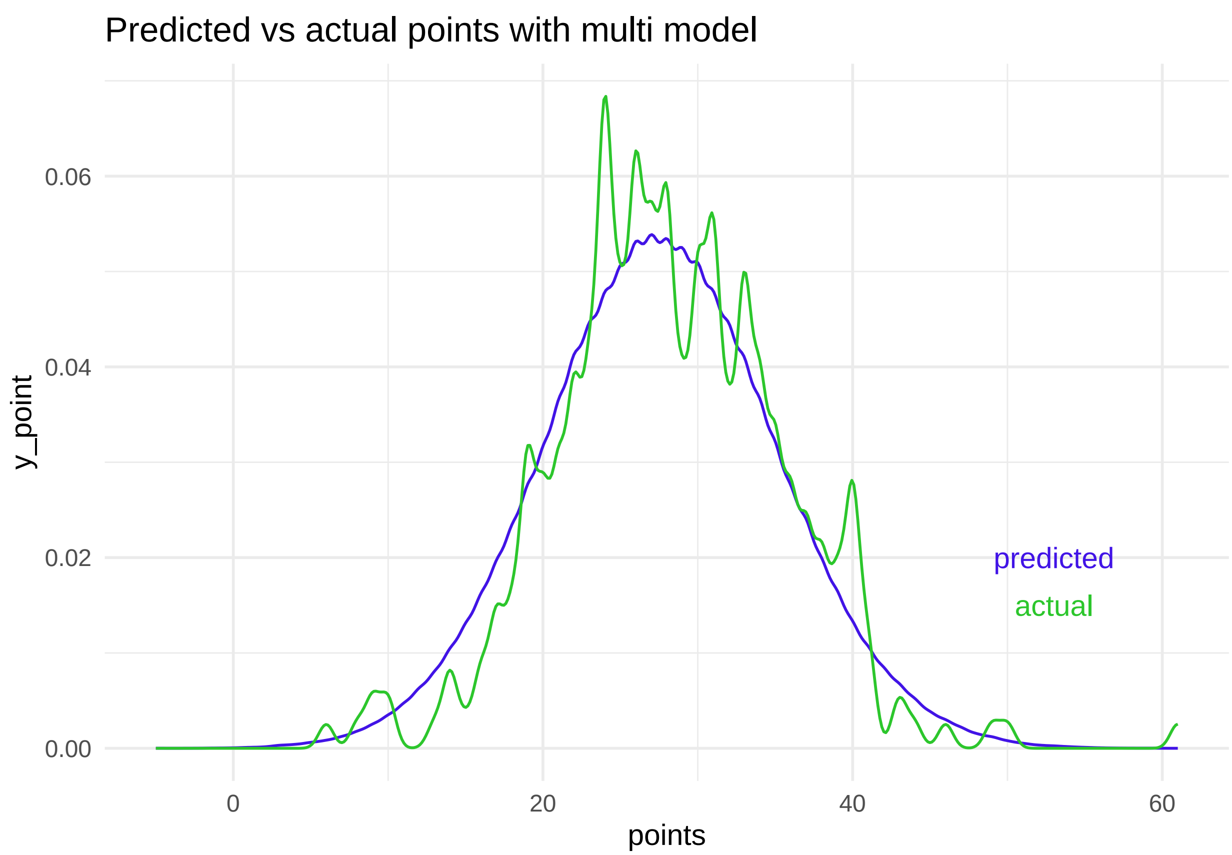
```
set.seed(123456)
# samples from posterior
post_samples_simple <- extract.samples(simple_model, n = 1000)

# prediction function
prediction_simple <- function(FGA, post_samples_simple) {
  mu <- post_samples_simple$a +
    post_samples_simple$b_1 * (FGA - FGA_mean)
  rnorm(1000, mu, post_samples_simple$sd)
}
```

```
# apply prediction fn to the test data
# create a separate df to draw plot easier

shaq_pred_vs_actual_simple <- shaq_test %>%
  rowwise() %>%
  mutate(PTS_pred = list(round(prediction_simple(FGA, post_samples_simple), 0))) %>%
  unnest(cols = c(PTS_pred)) %>%
  select(Season, PTS, PTS_pred)

head(shaq_pred_vs_actual_simple)
```

```
## # A tibble: 6 x 3
##   Season   PTS PTS_pred
##    <int> <int>    <dbl>
## 1      6    17       22
## 2      6    17       18
## 3      6    17       19
## 4      6    17       16
## 5      6    17       28
## 6      6    17       20
```

```
# points on the graph to put text
x_point <- max(shaq_pred_vs_actual_simple$PTS) - 8
y_point <- 0.02

ggplot() +
```

```
  geom_density(data = shaq_pred_vs_actual_simple, aes(x = PTS_pred), color = "#4113e6") +
  geom_text(aes(x = x_point, y = y_point, label = "predicted"), colour = "#4113e6") +
  geom_density(data = shaq_pred_vs_actual_simple, aes(x = PTS), color = "#2cc62c") +
  geom_text(aes(x = x_point, y = y_point - 0.005, label = "actual"), colour = "#2cc62c") +
  theme_minimal() +
  labs(title = "Predicted vs actual points with multi model",
       x = "points")
```



Task 2.2

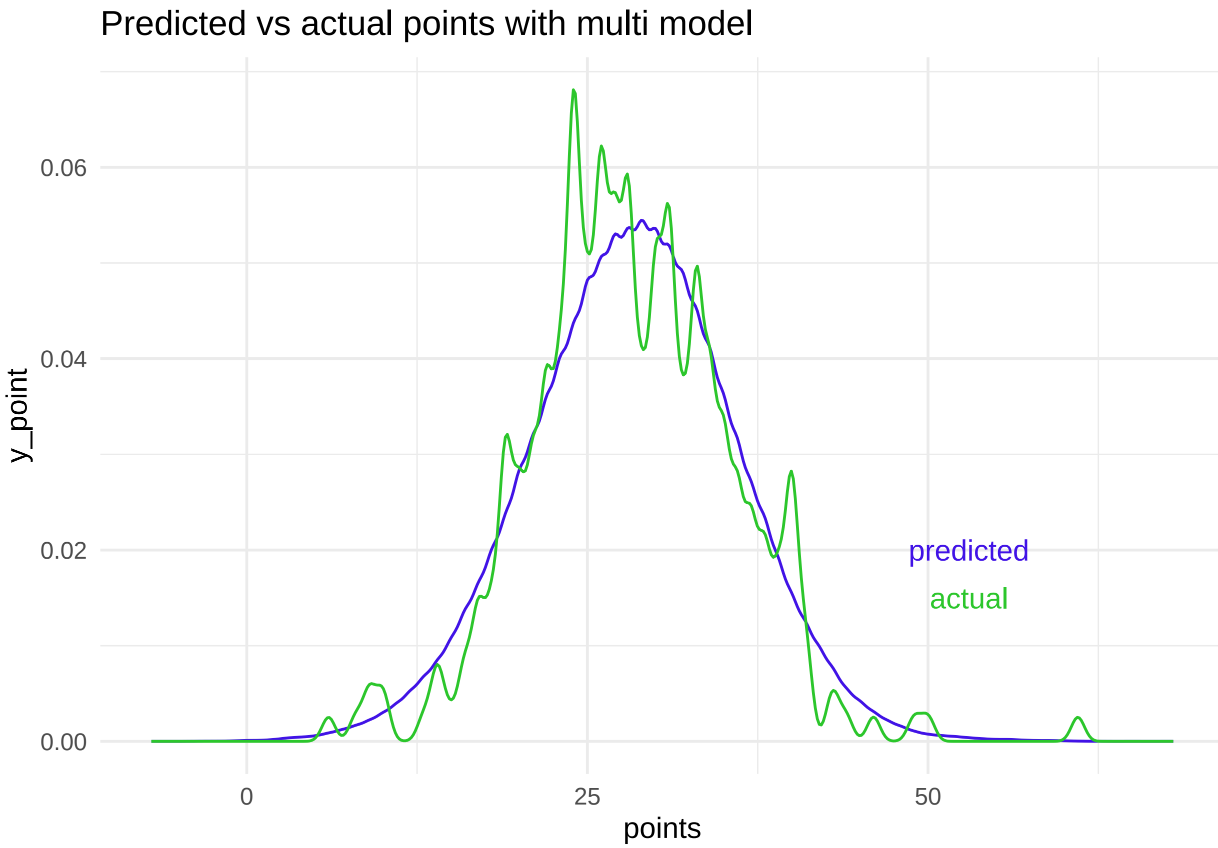
Use posterior samples from the multiple regression model that you estimated in Task 1.2 and the FGA and FTA data from the test set to predict new points. Create a plot that shows the predicted point distribution along the actual point distribution from Season `Season` 6 to 10.

```
# samples from posterior
post_samples_multi <- extract.samples(multi_model, n = 1000)

# prediction function
prediction_multi <- function(FGA, FTA, post_samples_multi) {
  mu <- post_samples_multi$a +
    post_samples_multi$b_1 * (FGA - FGA_mean) +
    post_samples_multi$b_2 * (FTA - FTA_mean)
  rnorm(1000, mu, post_samples_multi$sd)
}
```

```r
# apply prediction fn to the test data
# create a separate df to draw plot easier

shaq_pred_vs_actual_multi <- shaq_test %>%
  rowwise() %>%
  mutate(PTS_pred = list(round(prediction_multi(FGA, FTA, post_samples_multi), 0))) %>%
  unnest(cols = c(PTS_pred)) %>%
  select(Season, PTS, PTS_pred)

head(shaq_pred_vs_actual_multi)
```

```
## # A tibble: 6 x 3
##   Season   PTS PTS_pred
##    <int> <int>    <dbl>
## 1      6    17       25
## 2      6    17       25
## 3      6    17       24
## 4      6    17       18
## 5      6    17       12
## 6      6    17       20
```

```r
# points on the graph to put text
x_point <- max(shaq_pred_vs_actual_multi$PTS) - 8
y_point <- 0.02

ggplot() +
    geom_density(data = shaq_pred_vs_actual_multi, aes(x = PTS_pred), color = "#4113e6") +
    geom_text(aes(x = x_point, y = y_point, label = "predicted"), colour = "#4113e6") +
    geom_density(data = shaq_pred_vs_actual_multi, aes(x = PTS), color = "#2cc62c") +
    geom_text(aes(x = x_point, y = y_point - 0.005, label = "actual"), colour = "#2cc62c") +
    theme_minimal() +
    labs(title = "Predicted vs actual points with multi model",
         x = "points")
```


Predicted vs actual points with multi model
0.06
0.04
0.02
0.00
y_point
0
25
50
points
predicted
actual

Task Set 3

Task 3.1

Write a function `error()` that takes the predicted points $\hat{y}$ and the observed points y to compute the sum of squared errors:

$$\sum_i^n (\hat{y}_i - y_i)^2$$

Compute the squared errors for the simple regression model and the multiple regression model. Which model makes better predictions for the test data?

```
error <- function(pred, obs) {
  sum((pred - obs)^2)
}
```

```
# simple model
error(shaq_pred_vs_actual_simple$PTS_pred, shaq_pred_vs_actual_simple$PTS)
```

```
## [1] 16815670
```

```
# multi model
error(shaq_pred_vs_actual_multi$PTS_pred, shaq_pred_vs_actual_multi$PTS)
```

```
## [1] 11719708
```

Sum of squared errors is smaller for multiple regression model, therefore, we can say it makes better predictions

Task 3.2

For both models, compute the (non-squared) differences between each prediction and observation. Create a plot that shows the distributions of differences for both models.

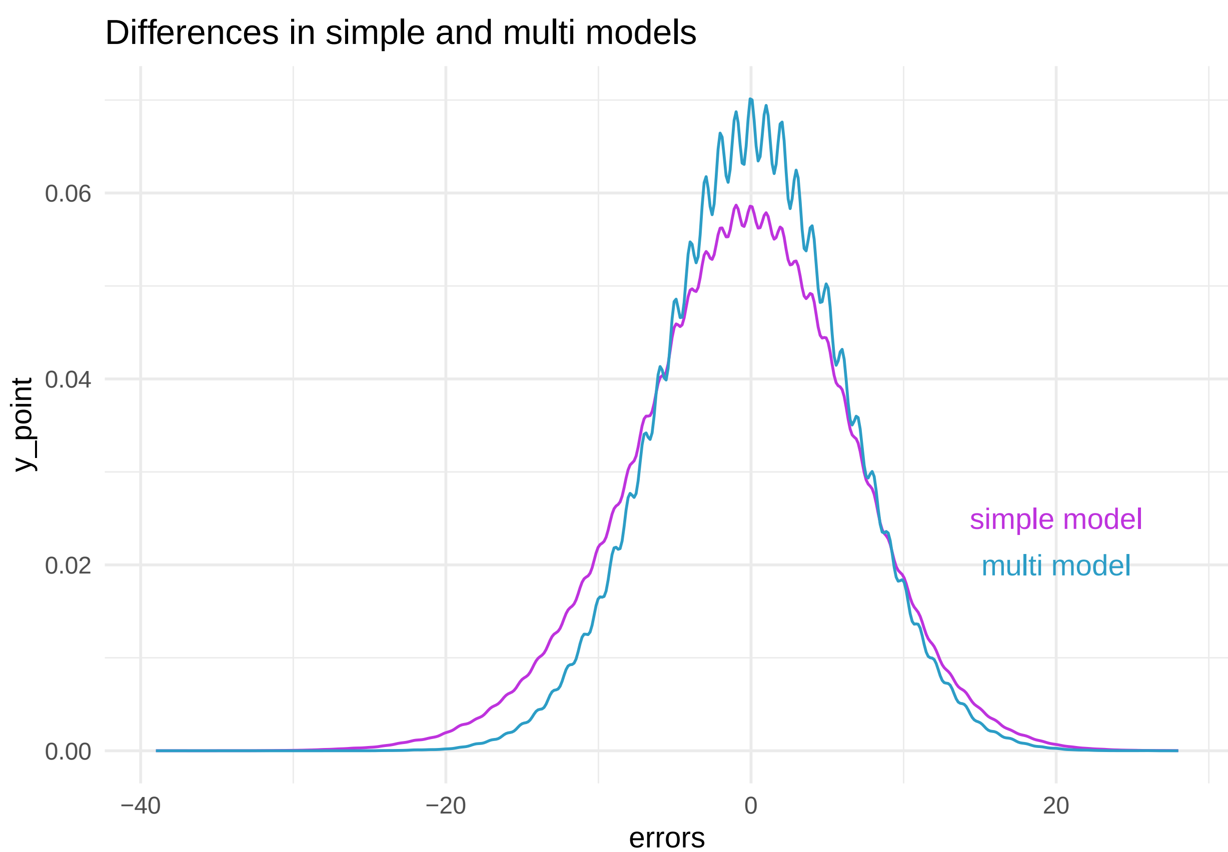
```
shaq_pred_vs_actual_simple <- shaq_pred_vs_actual_simple %>%
  mutate(diff = PTS_pred - PTS)

shaq_pred_vs_actual_multi <- shaq_pred_vs_actual_multi %>%
  mutate(diff = PTS_pred - PTS)

head(shaq_pred_vs_actual_simple)
```

```
## # A tibble: 6 x 4
##   Season   PTS PTS_pred  diff
##    <int> <int>    <dbl> <dbl>
## 1      6    17       22     5
## 2      6    17       18     1
## 3      6    17       19     2
## 4      6    17       16    -1
## 5      6    17       28    11
## 6      6    17       20     3
```

```r
head(shaq_pred_vs_actual_multi)
```

```
## # A tibble: 6 x 4
##   Season   PTS PTS_pred  diff
##    <int> <int>    <dbl> <dbl>
## 1      6    17       25     8
## 2      6    17       25     8
## 3      6    17       24     7
## 4      6    17       18     1
## 5      6    17       12    -5
## 6      6    17       20     3
```

```r
# points on the graph to put text
x_point <- max(shaq_pred_vs_actual_simple$diff) - 8
y_point <- 0.025

ggplot() +
    geom_density(data = shaq_pred_vs_actual_simple, aes(x = diff), color = "#be33dd") +
    geom_text(aes(x = x_point, y = y_point, label = "simple model"), colour = "#be33dd") +
    geom_density(data = shaq_pred_vs_actual_multi, aes(x = diff), color = "#2c9dc6") +
    geom_text(aes(x = x_point, y = y_point - 0.005, label = "multi model"), colour = "#2c9dc6") +
    theme_minimal() +
    labs(title = "Differences in simple and multi models",
         x = "errors")
```



Distribution of differences in multiple model is more narrow with more errors around zero, which also shows that it predicts better than the simple one.