Assignment 3

Kerem Karagöz Immanuel Klein

GitHub: https://github.com/immanuel-klein/bayesian-assignments.git

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.

```
#load packages here
library(dplyr)
library(ggplot2)
library(tinytex)
library(rethinking)
```

```
#load data here
shaq <- read.csv("shaq.csv")
```

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.first.seasons <- shaq %>% filter(Season >= 1 & Season <= 5)
head(shaq.first.seasons, n = 3)
```

```
  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
  FGA FG. X3P X3PA X3P. FT FTA   FT. ORB DRB TRB AST STL BLK TOV PF PTS GmSc
1   8 0.5   0    0   NA  4   7 0.571   5  13  18   2   1   3   8  6  12  8.3
2  16 0.5   0    0   NA  6  11 0.545   5  10  15   1   0   4   4  5  22 16.0
3  25 0.6   0    0   NA  5   8 0.625   4   9  13   1   1   3   4  4  35 26.0
  Pls.Mns
1      NA
2      NA
3      NA
```

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.

```
model1 <- quap(
  alist(
    PTS ~ dnorm(mu, sd),
    mu <- a + b * (FGA - mean(FGA)),
    # Because I have basically no knowledge of basketball metrics,
    # I choose the priors based on examples from the lecture code
    # and after some trial error with the quap function
    a ~ dnorm(20, 8),
    b ~ dunif(0, 3),
    sd ~ dunif(0, 8)
  ),
  data = shaq.first.seasons)
```

```r
summary(model1)
```

```
        mean         sd      5.5%     94.5%
a  27.029695 0.26744623 26.602264 27.457126
b   1.173326 0.05395668  1.087093  1.259559
sd  4.977557 0.18921863  4.675149  5.279965
```

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.

```r
model2 <- quap(
  alist(
    PTS ~ dnorm(mu, sd),
    mu <- a + b1 * (FGA - mean(FGA)) + b2 * (FTA - mean(FTA)),
    # Again: Because I have basically no knowledge of basketball metrics,
    # I choose the priors based on examples from the lecture code
    # and after some trial error with the quap function
    a ~ dnorm(20, 8),
    b1 ~ dunif(0, 3),
    b2 ~ dunif(0, 3),
    sd ~ dunif(0, 8)
  ),
  data = shaq.first.seasons)

summary(model2)
```

```
         mean         sd       5.5%      94.5%
a  27.0325065 0.23312993 26.6599198 27.4050931
b1  1.0493930 0.04849230  0.9718930  1.1268931
b2  0.6114339 0.05846217  0.5180001  0.7048678
sd  4.3383040 0.16488735  4.0747821  4.6018258
```

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)
head(shaq_test, n = 3)
```

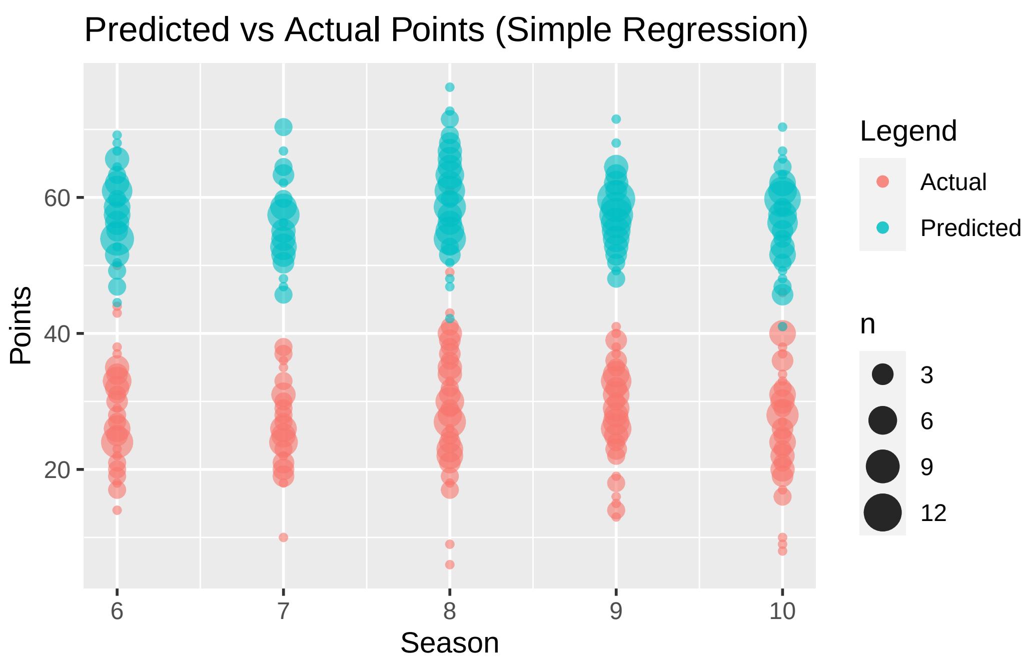
```
  Season SeasGm CarrGm  Date     Age  Tm Home Opp Win teamdiff GS Minutes FG
1      6      1    347 35741 25.6735 LAL    1 NYK   1        5  1      27  7
2      6      2    348 35743 25.6790 LAL    1 GSW   1       35  1      29 10
3      6      3    349 35745 25.6845 LAL    0 DAL   1       22  1      37 17
  FGA   FG. X3P X3PA X3P. FT FTA   FT. ORB DRB TRB AST STL BLK TOV PF PTS GmSc
1  12 0.583   0    0   NA  3   9 0.333   3   5   8   4   0   3   2  6  17 13.1
2  17 0.588   0    0   NA  7  12 0.583   4  15  19   3   0   2   2  3  27 24.7
3  23 0.739   0    0   NA  3   9 0.333   2  10  12   2   0   3   2  3  37 30.0
  Pls.Mns
1      NA
2      NA
3      NA
```

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.

```
# Extracting posterior samples
post.model1 <- extract.samples(model1)

# Replicating the mean values to match the dimensions
mean.FGA <- mean(shaq$FGA)
FGA.adj <- shaq_test$FGA - mean.FGA

# Making predictions
PTS.predicted <- post.model1$a + post.model1$a + post.model1$b %*% t(FGA.adj)

# Summarizing predictions
pred.summary <- apply(PTS.predicted, 2, mean)
actual.pts <- shaq_test$PTS
```

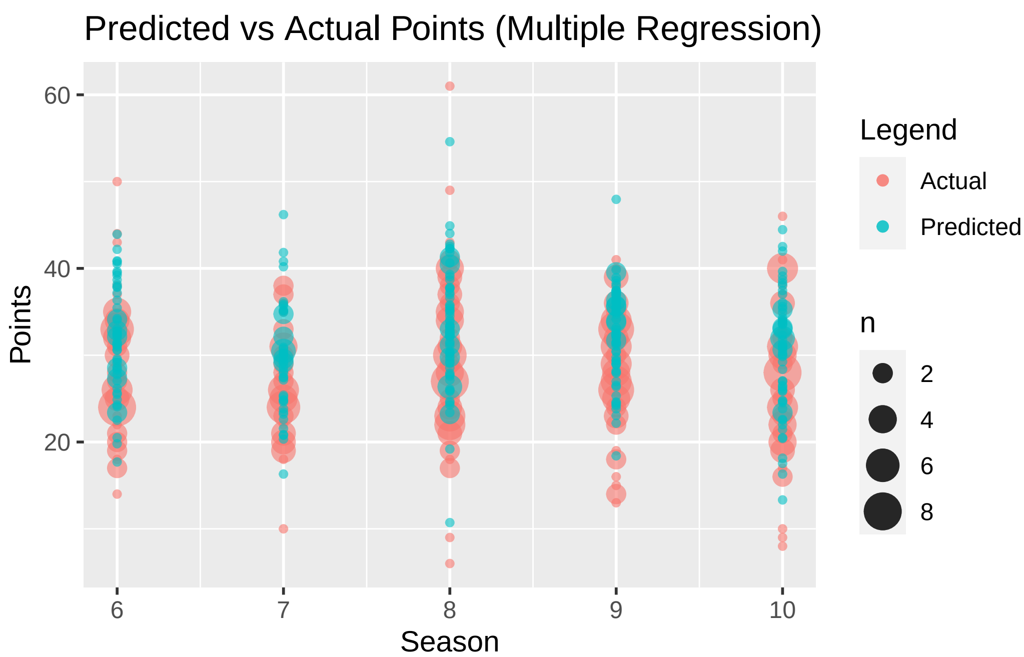
```
# Plotting predictions vs actual
plot.df1 <- data.frame(
  Season = shaq_test$Season,
  Actual = actual.pts,
  Predicted = pred.summary
)

ggplot(plot.df1, aes(x = Season)) +
  geom_count(aes(y = Actual, color = 'Actual'), alpha = 0.6) +
  geom_count(aes(y = Predicted, color = 'Predicted'), alpha = 0.6) +
  labs(title = 'Predicted vs Actual Points (Simple Regression)',
       y = 'Points', color = 'Legend')
```



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.

```r
# Extracting posterior samples
post.model2 <- extract.samples(model2)

# Replicating the mean values to match the dimensions for FTA as well
mean.FTA <- mean(shaq$FTA)
FTA.adj <- shaq_test$FTA - mean.FTA

# Making predictions
PTS.predicted.multi <- post.model2$a + post.model2$b1 %*% t(FGA.adj) +
post.model2$b2 %*% t(FTA.adj)

# Summarizing predictions
pred.summary.multi <- apply(PTS.predicted.multi, 2, mean)
actual.pts.multi <- shaq_test$PTS

# Plotting predictions vs actual
plot.df2 <- data.frame(
  Season = shaq_test$Season,
  Actual = actual.pts.multi,
  Predicted = pred.summary.multi
)

ggplot(plot.df2, aes(x = Season)) +
  geom_count(aes(y = Actual, color = 'Actual'), alpha = 0.6) +
  geom_count(aes(y = Predicted, color = 'Predicted'), alpha = 0.6) +
  labs(title = 'Predicted vs Actual Points (Multiple Regression)',
       y = 'Points', color = 'Legend')
```



Predicted vs Actual Points (Multiple Regression)
Points
60
40
20
6
7
8
9
10
Season
Legend
Actual
Predicted
n
2
4
6
8

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?

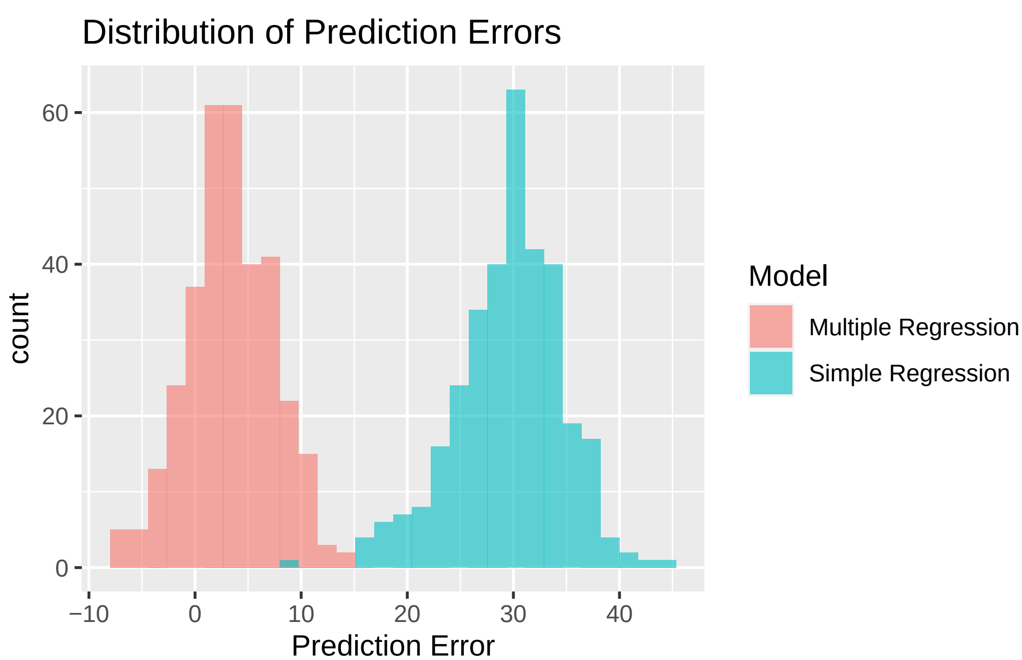
```r
# Function to compute sum of squared errors
error <- function(actual, predicted) {
  sum((predicted - actual)^2)
}

# Calculate SSE for both models
sse <- sapply(list(
  simple = list(actual = actual.pts, predicted = pred.summary),
  multiple = list(actual = actual.pts.multi, predicted = pred.summary.multi)
), function(x) error(x$actual, x$predicted))

# Determine the better model
better.model <- ifelse(sse["simple"] < sse["multiple"],
                       "Simple Regression", "Multiple Regression")

cat("Sum of Squared Errors (SSE):\n",
    "Simple Regression: ", sse["simple"], "\n",
    "Multiple Regression: ", sse["multiple"], "\n",
    "Better Model: ", better.model, "\n")
```

```
Sum of Squared Errors (SSE):
 Simple Regression:  296260.4
 Multiple Regression:  9043.156
 Better Model:  Multiple Regression
```

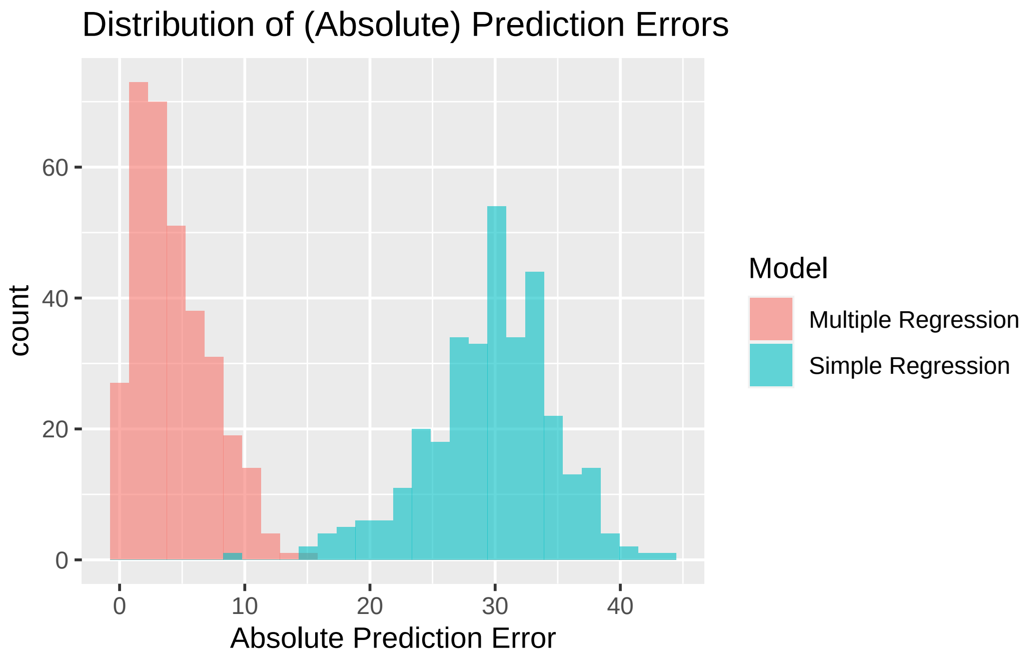
Task 3.2

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.

```
# Calculate differences for both models
differences <- data.frame(
  Model = rep(c("Simple Regression", "Multiple Regression"),
              each = length(actual.pts)),
  Differences = c(pred.summary - actual.pts,
                  pred.summary.multi - actual.pts.multi)
)

# Calculate absolute differences for both models
differences.abs <- data.frame(
  Model = rep(c("Simple Regression", "Multiple Regression"),
              each = length(actual.pts)),
  Differences = c(abs(pred.summary - actual.pts),
                  abs(pred.summary.multi - actual.pts.multi))
)

# Plot distributions of differences
ggplot(differences, aes(x = Differences, fill = Model)) +
  geom_histogram(alpha = 0.6, position = "identity", bins = 30) +
  labs(title = "Distribution of Prediction Errors",
       x = "Prediction Error", fill = "Model")
```



```r
# Plot distributions of absolute differences
ggplot(differences.abs, aes(x = Differences, fill = Model)) +
  geom_histogram(alpha = 0.6, position = "identity", bins = 30) +
  labs(title = "Distribution of (Absolute) Prediction Errors",
       x = "Absolute Prediction Error", fill = "Model")
```



Remark:

We're plotting both the differences and their absolute values: The absolute plot represents the smaller error of the multiple regression better. The standard plot shows that for the simple regression differences are positive, indicating that prediction is higher than the actual point, the model overshoots.