Assignment-3 Q&A Bilge Salman (03796071), Salim Kaplan (0378856) GitHub link: https://github.com/salimkaplan/BayesIntro24-Assignments/tree/main

```r
# load packages here
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
library(tibble)
library(rethinking)
```

```
Zorunlu paket yükleniyor: cmdstanr
This is cmdstanr version 0.8.0
- CmdStanR documentation and vignettes: mc-stan.org/cmdstanr
- CmdStan path: C:/Users/DeLL/.cmdstan/cmdstan-2.35.0
- CmdStan version: 2.35.0
Zorunlu paket yükleniyor: posterior
This is posterior version 1.5.0
```

```
Attaching package: 'posterior'

The following objects are masked from 'package:stats':

    mad, sd, var

The following objects are masked from 'package:base':

    %in%, match

Zorunlu paket yükleniyor: parallel
rethinking (Version 2.40)

Attaching package: 'rethinking'

The following object is masked from 'package:purrr':

    map

The following object is masked from 'package:stats':

    rstudent
```

```r
library(here)
```

```
here() starts at C:/Users/DeLL/Desktop/Introduction to Bayesian Data Analysis/BayesIntro24-As
```

```r
library(rstan)
```

```
Zorunlu paket yükleniyor: StanHeaders

rstan version 2.32.6 (Stan version 2.32.2)

For execution on a local, multicore CPU with excess RAM we recommend calling
options(mc.cores = parallel::detectCores()).
To avoid recompilation of unchanged Stan programs, we recommend calling
rstan_options(auto_write = TRUE)
For within-chain threading using `reduce_sum()` or `map_rect()` Stan functions,
change `threads_per_chain` option:
```

```
rstan_options(threads_per_chain = 1)

Do not specify '-march=native' in 'LOCAL_CPPFLAGS' or a Makevars file

Attaching package: 'rstan'

The following objects are masked from 'package:rethinking':

    stan, traceplot

The following objects are masked from 'package:posterior':

    ess_bulk, ess_tail

The following object is masked from 'package:tidyr':

    extract
```

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

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.

```r
# Task 1.1
# Loading Shaq data
data_path <- here("shaq.csv")
shaq <- read.csv(data_path)

# Training data set
shaq_training <- shaq %>% filter(Season <= 5)

# simple linear model
model_1.1 <- quap(
  alist(
    PTS ~ dnorm(mu, sigma), # likelihood
    mu <- a + b * FGA, # linear model
    a ~ dunif(0,10), # prior intercept
    b ~ dunif(0, 3), # prior rate of change (slope)
    sigma ~ dunif(0,10) # prior sd
  ),
  data = shaq_training
)

# Summary table of posterior distribution
precis(model_1.1)
```

```
          mean         sd     5.5%    94.5%
a     4.951224 1.05032506 3.272601 6.629846
b     1.173327 0.05395672 1.087093 1.259560
sigma 4.977561 0.18921885 4.675153 5.279969
```

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.

```
# Task 1.2
# multiple linear model
model_1.2 <- quap(
  alist(
    PTS ~ dnorm(mu, sigma),
    mu <- a + bFGA * FGA * 2 + bFTA * FTA,
    a ~ dunif(0, 10),
    bFGA ~ dunif(0, 1),
    bFTA ~  dunif(0, 1),
    sigma ~ dunif(0, 8)
  ),
  data = shaq_training
)

# Summary table of posterior distribution
precis(model_1.2)
```

```
           mean         sd       5.5%     94.5%
a     1.2451791 0.98173793 -0.3238277 2.8141859
bFGA  0.5247975 0.02424912  0.4860427 0.5635523
```

```
bFTA  0.6114655 0.05846934  0.5180202 0.7049108
sigma 4.3388361 0.16493786  4.0752335 4.6024387
```

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.

```r
# Test data set
shaq_test <- shaq %>% filter(Season >= 6 & Season <= 10)
```

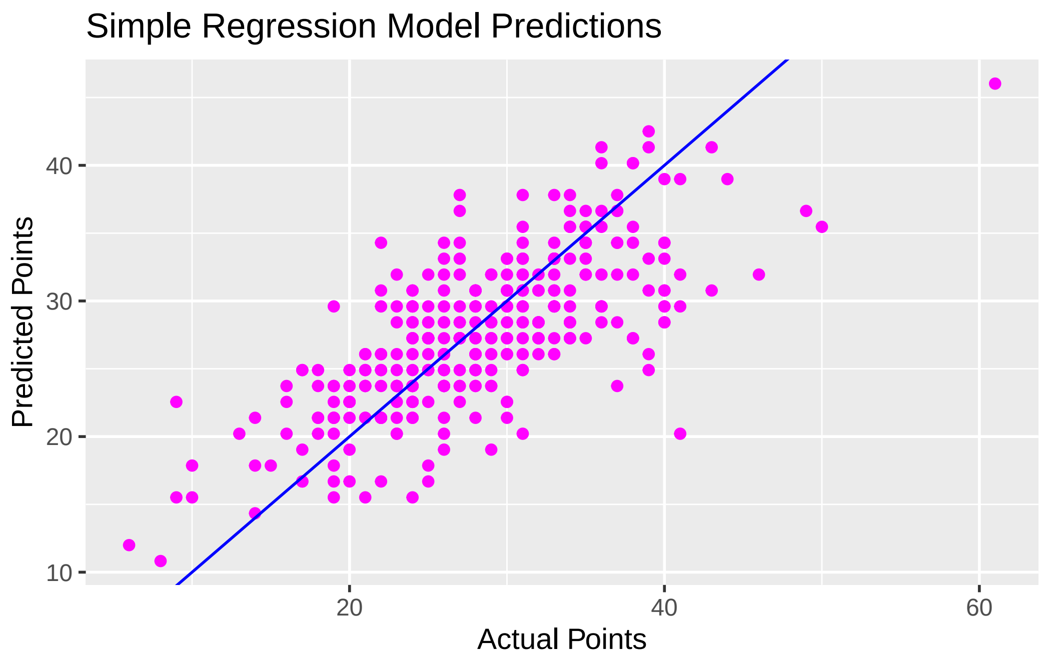
Task 2.1

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.

```r
# Making predictions using the Task 1.1 model
predictions_1.1 <- link(model_1.1, data = shaq_test)

# Getting average estimates
pred_mean_1.1 <- apply(predictions_1.1, 2, mean)

# Plot that shows the predicted point distribution along the actual point distribution
ggplot() +
  geom_point(aes(x = shaq_test$PTS, y = pred_mean_1.1), color = "#ff02ff") +
  geom_abline(intercept = 0, slope = 1, color = "blue") +
  labs(x = "Actual Points", y = "Predicted Points", title = "Simple Regression Model Predi
```



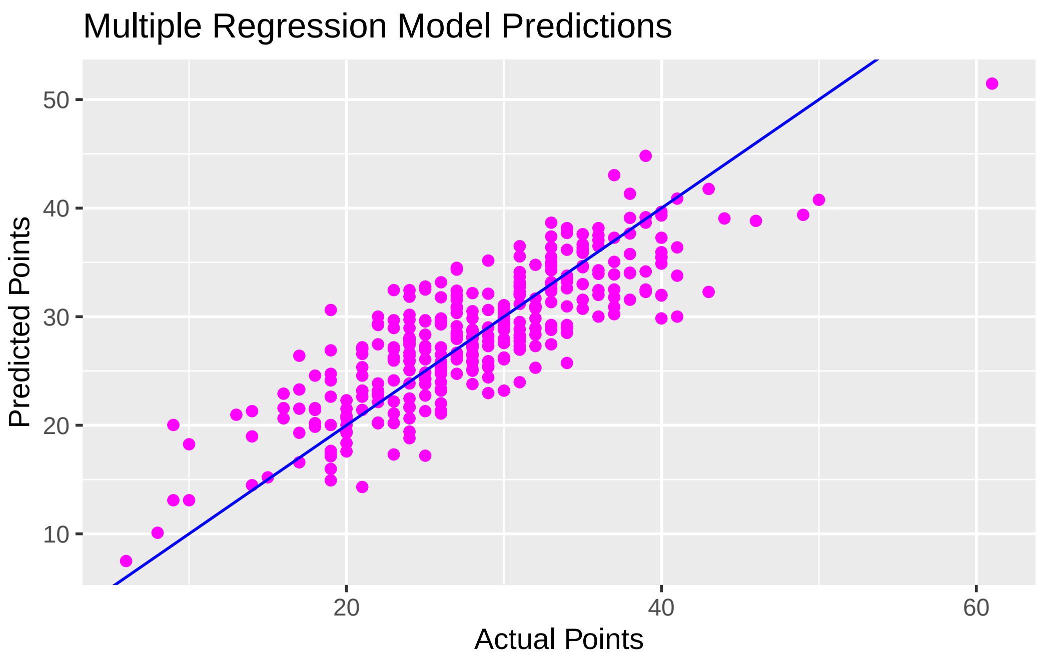
Task 2.2

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
#  Making predictions using the Task 1.2 model
predictions_1.2 <- link(model_1.2, data = shaq_test)

# Getting average estimates
pred_mean_1.2 <- apply(predictions_1.2, 2, mean)

# Plot that shows the predicted point distribution along the actual point distribution
ggplot() +
  geom_point(aes(x = shaq_test$PTS, y = pred_mean_1.2), color = "#ff02ff") +
  geom_abline(intercept = 0, slope = 1, color = "blue") +
  labs(x = "Actual Points", y = "Predicted Points", title = "Multiple Regression Model Pre
```


Multiple Regression Model Predictions
Predicted Points
Actual Points
10
20
30
40
50
20
40
60

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
# Error function to calculate the sum of squared errors
error <- function(predicted, observed) {
  sum((predicted - observed)^2)
}

# Error for simple regression model (Task 1.1)
error_1.1 <- error(pred_mean_1.1, shaq_test$PTS)

# Error for multiple regression model (Task 1.2)
error_1.2 <- error(pred_mean_1.2, shaq_test$PTS)

# Results
cat( "Error for simple regression model", error_1.1, "\n")
```

```
Error for simple regression model 8557.109
```

```r
cat("Error for multiple regression model", error_1.2, "\n")
```

```
Error for multiple regression model 5417.688
```

```r
# Which model makes better predictions for the test data?

# The Multiple Regression Model looks at not only FGA but also FTA.
# This additional information helps the model make more accurate predictions.
# Therefore, a multiple regression model that includes more variables and relies
# on more information has the potential to improve predictions.
```

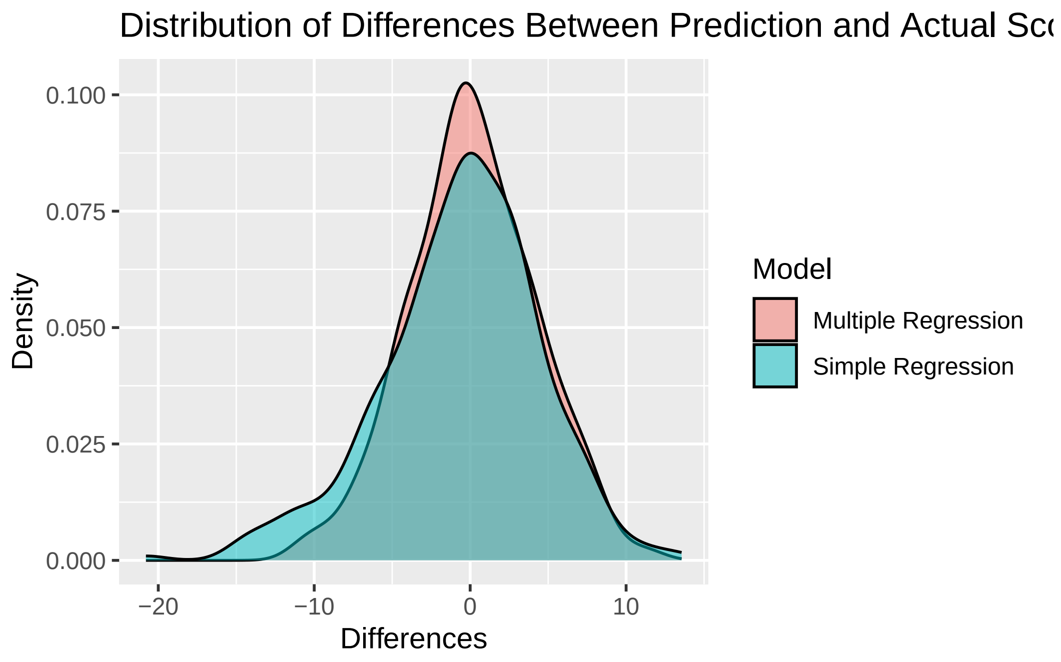
Task 3.2

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.

```r
# Computing the (non-squared) differences
diff_1.1 <- pred_mean_1.1 - shaq_test$PTS
diff_1.2 <- pred_mean_1.2 - shaq_test$PTS

# Plot shows the distributions of differences for both models
data_diff <- data.frame(
  Model = rep(c("Simple Regression", "Multiple Regression"),
              each = length(diff_1.1)),
  Difference = c(diff_1.1, diff_1.2)
)

ggplot(data_diff, aes(x = Difference, fill = Model)) +
  geom_density(alpha = 0.5) +
  labs(title = "Distribution of Differences Between Prediction and Actual Scores",
       x = "Differences", y = "Density")
```


Distribution of Differences Between Prediction and Actual Sco
Density
Differences
Model
Multiple Regression
Simple Regression