DATA 612: Project 1 - Global Baseline Predictors and RMSE

Derek G Nokes

2019-06-09

Introduction

In the first section of this project, we implement the required functionality and validate it using the data from parts K through P of the Coursera/Stanford Networks Illustrated course. In the second section of the project, we apply the implemented functionality to our own data.

Code Validation

If you choose to work with a large dataset, you're encouraged to also create a small, relatively dense "user-item" matrix as a subset so that you can hand-verify your calculations.

Prior to using our own data, we reproduce the data used in parts K through P of the Coursera/Stanford Networks Illustrated course and use it to validate our work.

First we create the data used in the video series:

```
# create validation data
data<-matrix(c(5,NA,4,NA,4,
               4,3,5,3,4,
               4,2,NA,NA,3,
               2,2,3,1,2,
               4,NA,5,4,5,
               4,2,5,4,4),
             nrow=6,byrow=T)
# create split index
splitIndex<-matrix(c(T,T,T,T,T,
              T,T,T,F,T,
              F,T,T,T,T,
              T,F,T,T,T,
              T,T,F,T,T,
              T,T,T,T,F),byrow=T,nrow=6)
# define user and item labels
users<-c('A','B','C','D','E','F')
items<-c('I','II','III','IV','V')
# label row and column names
rownames(data)<-users
colnames(data)<-items
```

```
# create data frame
data_df<-data.frame(users,data)
```

Create data used in parts K through P of the Coursera/Stanford Networks Illustrated course.

Table 1: Coursera/Stanford Networks Illustrated Data

	I	II	III	IV	V
A	5	NA	4	NA	4
B	4	3	5	3	4
C	4	2	NA	NA	3
D	2	2	3	1	2
E	4	NA	5	4	5
F	4	2	5	4	4

Next we split the data:

```
# convert data to long
long_data_df <- data_df %>%
  gather(key=item,value=rating,-users)
# convert split index to long
longSplitIndex<-data.frame(users,splitIndex) %>%
  gather(key=item,value=include,-users)
# extract training set
train<-long_data_df[longSplitIndex$include,]
# extract testing set
test<-long_data_df[!longSplitIndex$include,]
```

Split the data into training and testing data sets.

Now we compute the mean rating for the training set:

```
# compute mean rating for training and testing data sets
meanRatingTrain<-mean(train$rating,na.rm=T)
meanRatingTest<-mean(test$rating,na.rm=T)
```

Compute the mean rating for the training and testing data sets.

The mean rating for the training set is 3.5, which matches value determined in the Coursera/Stanford Networks Illustrated course videos.

Next, we define a function to compute the root mean squared error (RMSE) as follows:

```
RMSE <- function(test,train){
  e<-(test-mean(train,na.rm=T))
  rmse<-sqrt(mean(e^2,na.rm=T))
  return(rmse)
}
```

Then we compute the root mean squared error (RMSE) as follows:

```
# compute root mean squared error (RMSE) of training and testing data sets
rmseTrain<-RMSE(train$rating,train$rating)
rmseTest<-RMSE(test$rating,train$rating)
```

The training and testing set root mean squared errors (RMSEs) are 1.1619 and 1.0247 respectively, which again matches the values determined in the Coursera/Stanford Networks Illustrated course videos.

Now, we compute the user and item biases:

```
# create user biases
userBias <- train %>% filter(!is.na(rating)) %>%
  group_by(users) %>%
  summarise(sum = sum(rating), count = n()) %>%
  mutate(bias = sum/count-meanRatingTrain) %>%
  select(users, userBias = bias)
# create item biases
itemBias <- train %>% filter(!is.na(rating)) %>%
  group_by(item) %>%
  summarise(sum = sum(rating), count = n()) %>%
  mutate(bias = sum/count-meanRatingTrain) %>%
  select(item, itemBias = bias)
```

Table 2: Coursera/Stanford Networks Illustrated User Biases

users	userBias
A	0.8333333
B	0.5000000
C	-1.0000000
D	-1.5000000
E	0.8333333
F	0.2500000

Table 3: Coursera/Stanford Networks Illustrated Item Biases

item	itemBias
I	0.300000
II	-1.166667
III	0.750000
IV	-0.500000
V	0.100000

Both the user and item biases shown above match the results from the Coursera/Stanford Networks Illustrated videos.

Finally, we create the training and testing set baseline predictions:

```
# create training set baseline table
train_table <- train %>% left_join(userBias, by = "users") %>%
  left_join(itemBias, by = "item") %>%
  mutate(meanRating = meanRatingTrain) %>%
  mutate(baseline = meanRating + userBias + itemBias)
# create testing set baseline table
test_table <- test %>% left_join(userBias, by = "users") %>%
  left_join(itemBias, by = "item") %>%
  mutate(meanRating = meanRatingTrain) %>%
  mutate(baseline = meanRating + userBias + itemBias)
```

Table 4: Coursera/Stanford Networks Illustrated Training Set Baseline Predictions

users	item	rating	userBias	itemBias	meanRating	baseline
A	I	5	0.8333333	0.300000	3.5	4.633333
B	I	4	0.5000000	0.300000	3.5	4.300000
D	I	2	-1.5000000	0.300000	3.5	2.300000
E	I	4	0.8333333	0.300000	3.5	4.633333
F	I	4	0.2500000	0.300000	3.5	4.050000
A	II	NA	0.8333333	-1.166667	3.5	3.166667
B	II	3	0.5000000	-1.166667	3.5	2.833333
C	II	2	-1.0000000	-1.166667	3.5	1.333333
E	II	NA	0.8333333	-1.166667	3.5	3.166667
F	II	2	0.2500000	-1.166667	3.5	2.583333
A	III	4	0.8333333	0.750000	3.5	5.083333
B	III	5	0.5000000	0.750000	3.5	4.750000
C	III	NA	-1.0000000	0.750000	3.5	3.250000
D	III	3	-1.5000000	0.750000	3.5	2.750000
F	III	5	0.2500000	0.750000	3.5	4.500000
A	IV	NA	0.8333333	-0.500000	3.5	3.833333
C	IV	NA	-1.0000000	-0.500000	3.5	2.000000
D	IV	1	-1.5000000	-0.500000	3.5	1.500000
E	IV	4	0.8333333	-0.500000	3.5	3.833333
F	IV	4	0.2500000	-0.500000	3.5	3.250000
A	V	4	0.8333333	0.100000	3.5	4.433333
B	V	4	0.5000000	0.100000	3.5	4.100000
C	V	3	-1.0000000	0.100000	3.5	2.600000
D	V	2	-1.5000000	0.100000	3.5	2.100000
E	V	5	0.8333333	0.100000	3.5	4.433333

Table 5: Coursera/Stanford Networks Illustrated Testing Set Baseline Predictions

users	item	rating	userBias	itemBias	meanRating	baseline
C	I	4	-1.0000000	0.300000	3.5	2.8000000
D	II	2	-1.5000000	-1.166667	3.5	0.8333333
E	III	5	0.8333333	0.750000	3.5	5.0833333
B	IV	3	0.5000000	-0.500000	3.5	3.5000000
F	V	4	0.2500000	0.100000	3.5	3.8500000

We now combine the training and testing baseline predictions into a matrix to compare against the results from the Coursera/Stanford Networks Illustrated video series:

```
# create baseline matrix to compare against video example
baselineTable<-rbind(train_table[,c('users','item','baseline')],
  test_table[,c('users','item','baseline')]) %>%
  spread(key=item,value=baseline) %>% column_to_rownames('users')
```

Table 6: Coursera/Stanford Networks Illustrated Baseline Prediction Matrix

	I	II	III	IV	V
A	4.633333	3.1666667	5.083333	3.833333	4.433333
B	4.300000	2.8333333	4.750000	3.500000	4.100000
C	2.800000	1.3333333	3.250000	2.000000	2.600000
D	2.300000	0.8333333	2.750000	1.500000	2.100000
E	4.633333	3.1666667	5.083333	3.833333	4.433333
F	4.050000	2.5833333	4.500000	3.250000	3.850000

The results directly above almost match those from the Coursera/Stanford Networks Illustrated video series. There are a few places in the video on the baseline prediction and RMSE where the results are rounded to make the arithmetic easier to do by hand and this results in a minor difference in the baseline prediction and the root mean squared error.

Finally, we create a summary table to compare the root mean squared error (RMSE) of the (raw) mean rating and the baseline predictions for both the training and testing data sets:

```
RMSE_baseline <- function(y,yBaseline){
  e<-(y-yBaseline)
  rmse<-sqrt(mean(e^2,na.rm=T))
  return(rmse)
}
# compute baseline RMSE for training set
rmseTrainBaseline<-RMSE_baseline(train_table$rating,train_table$baseline)
# compute baseline RMSE for testing set
```

```
rmseTestBaseline<-RMSE_baseline(test_table$rating,test_table$baseline)

rmses<-matrix(c(rmseTrain,rmseTrainBaseline,rmseTest,rmseTestBaseline),
  byrow=T,nrow=2)
setLabels<-c('Train','Test')
rmseTypeLabels<-c('Mean (Raw)','Baseline')
colnames(rmses)<-rmseTypeLabels
rownames(rmses)<-setLabels
```

Table 7: Coursera/Stanford Networks Illustrated RMSE Summary

	Mean (Raw)	Baseline
Train	1.161895	0.4800174
Test	1.024695	0.7849275

As in the video, we see a significant improvement in the root mean squared error (RMSE) for the baseline prediction when compared to the (raw) average prediction.

Application

In this section, we apply the implemented functionality to our own data set.

Briefly describe the recommender system that you're going to build out from a business perspective, e.g. "This system recommends data science books to readers."

The system to be built out is intended to provide managed account program recommendations to investors on an online platform. The system is initially to be based on explicit program ratings provided by platform users. Eventually the business expects to expand the recommender system to use managed account transaction data and program characteristics to improve the utility of the recommendations. The inventory of programs is currently relatively small relative to the number of users, so the ratings data is relatively dense.

Find a dataset, or build out your own toy dataset. As a minimum requirement for complexity, please include numeric ratings for at least five users, across at least five items, with some missing data. Load your data into (for example) an R or pandas dataframe, a Python dictionary or list of lists, (or another data structure of your choosing). From there, create a user-item matrix.

Since we don't have the required data, we build a toy data set:

```
# define random seed
randomSeed<-1234567
# set seed
set.seed(randomSeed)
```

```
# define, number of users, number of items
nUsers<-30
nItems<-15
# define highest and lowest ratings
lowestRating<-1
highestRating<-5
# define probability of a rating (including NA)
ratingProbabilities<-c(0.05,0.1,0.4,0.25,0.1,0.1)
# define rating scale including NA
ratingScale<-c(seq(lowestRating,highestRating),NA)
# create random user x item matrix of ratings by sampling from ratings scale
toyRatingsDf<-matrix(sample(ratingScale,
  size=nUsers*nItems,replace=TRUE,prob=ratingProbabilities),
  nUsers,nItems)
# create user labels
users <- paste("User_", seq(1,nUsers), sep="")
# create item labels
items <- paste("Item_", seq(1,nItems), sep="")
# assign row names
rownames(toyRatingsDf)<-users
# assign column names
colnames(toyRatingsDf)<-items
# convert matrix to data frame
toyRatingsDf <- data.frame(toyRatingsDf)
```

Break your ratings into separate training and test datasets.

We convert to long

```
longToyRatingsDf <- toyRatingsDf  %>%
  rownames_to_column(var='users') %>%
  gather(key = item, value = rating, -users)
```

We split the data by sub-selecting a random set of users and items:

```
# define random seed
randomSeed<-1234567
# set seed
set.seed(randomSeed)
# define train/test split ratio
splitRatio<-0.75
# create boolean train/test split index
splitIndex<-sample(c(T,F),prob=c(splitRatio,1-splitRatio),
  replace=T,size=nUsers*nItems)
# extract training samples
```

```
longTrainToyRatingDf<-longToyRatingsDf[splitIndex,]
# extract testing samples
longTestToyRatingDf<-longToyRatingsDf[!splitIndex,]
# convert train and test data sets from 'long' to 'wide'
trainToyRatingDf<-train %>% spread(key=item,value=rating)
testToyRatingDf<-test %>% spread(key=item,value=rating)
```

We use 75% of the data for training and use the remaining 25% for testing.

Using your training data, calculate the raw average (mean) rating for every user-item combination.

We compute the (raw) mean rating for the training set:

```
# compute mean rating for training and testing data sets
meanRatingTrain<-mean(longTrainToyRatingDf$rating,na.rm=T)
```

Calculate the RMSE for raw average for both your training data and your test data.

We calculate the root mean squared error (RMSE) for both our training and testing data:

```
# compute root mean squared error (RMSE) of training and testing data sets
rmseTrain<-RMSE(longTrainToyRatingDf$rating,longTrainToyRatingDf$rating)
rmseTest<-RMSE(longTestToyRatingDf$rating,longTrainToyRatingDf$rating)
```

The training and testing set root mean squared errors (RMSEs) are 0.4944 and 1.7666 respectively.

Using your training data, calculate the bias for each user and each item.

We compute the user and item biases:

```
# create user biases
userBias <- longTrainToyRatingDf %>%
  filter(!is.na(rating)) %>%
  group_by(users) %>%
  summarise(sum = sum(rating), count = n()) %>%
  mutate(bias = sum/count-meanRatingTrain) %>%
  select(users, userBias = bias)
# create item biases
itemBias <- longTrainToyRatingDf %>%
  filter(!is.na(rating)) %>%
  group_by(item) %>%
  summarise(sum = sum(rating), count = n()) %>%
  mutate(bias = sum/count-meanRatingTrain) %>%
  select(item, itemBias = bias)
```

From the raw average, and the appropriate user and item biases, calculate the baseline predictors for every user-item combination.

Table 8: Toy Training Data - User Biases

users	userBias
User_1	0.2747508
User_10	0.0747508
User_11	0.0747508
User_12	-0.3252492
User_13	-0.3002492
User_14	0.3525286
User_15	-0.3141381
User_16	0.0292963
User_17	-0.1752492
User_18	0.0747508
User_19	-0.1752492
User_2	0.0191953
User_20	-0.1175569
User_21	-0.0252492
User_22	0.0191953
User_23	0.0191953
User_24	0.0747508
User_25	-0.1752492
User_26	-0.0406338
User_27	-0.2434310
User_28	-0.0252492
User_29	0.0747508
User_3	0.0292963
User_30	0.0191953
User_4	0.0191953
User_5	-0.0252492
User_6	0.2414175
User_7	0.0747508
User_8	0.2414175
User_9	0.3020236

Table 9: Toy Training Data - Item Biases

item	itemBias
Item_1	0.0484350
Item_10	-0.0134845
Item_11	-0.0363603
Item_12	0.1997508
Item_13	-0.0339448
Item_14	-0.1311315
Item_15	0.0247508
Item_2	-0.0919158
Item_3	-0.0502492
Item_4	-0.0041965
Item_5	0.0033223
Item_6	0.0747508
Item_7	-0.0568281
Item_8	-0.0252492
Item_9	0.0509413

We create the training and testing set baseline predictions:

```
# create training set baseline table
train_table <- longTrainToyRatingDf %>%
  left_join(userBias, by = "users") %>%
  left_join(itemBias, by = "item") %>%
  mutate(meanRating = meanRatingTrain) %>%
  mutate(baseline = meanRating + userBias + itemBias)
# create testing set baseline table
test_table <- longTestToyRatingDf %>%
  left_join(userBias, by = "users") %>%
  left_join(itemBias, by = "item") %>%
  mutate(meanRating = meanRatingTrain) %>%
  mutate(baseline = meanRating + userBias + itemBias)
```

Table 10: Baseline Predictions (Training)

users	item	rating	userBias	itemBias	meanRating	baseline
User_1	Item_1	4	0.2747508	0.0484350	3.425249	3.748435
User_2	Item_1	NA	0.0191953	0.0484350	3.425249	3.492879
User_4	Item_1	3	0.0191953	0.0484350	3.425249	3.492879
User_6	Item_1	4	0.2414175	0.0484350	3.425249	3.715102
User_7	Item_1	3	0.0747508	0.0484350	3.425249	3.548435
User_9	Item_1	4	0.3020236	0.0484350	3.425249	3.775708

Table 10: Baseline Predictions (Training) *(continued)*

users	item	rating	userBias	itemBias	meanRating	baseline
User_11	Item_1	4	0.0747508	0.0484350	3.425249	3.548435
User_12	Item_1	3	-0.3252492	0.0484350	3.425249	3.148435
User_13	Item_1	3	-0.3002492	0.0484350	3.425249	3.173435
User_14	Item_1	4	0.3525286	0.0484350	3.425249	3.826213
User_16	Item_1	4	0.0292963	0.0484350	3.425249	3.502981
User_17	Item_1	3	-0.1752492	0.0484350	3.425249	3.298435
User_19	Item_1	4	-0.1752492	0.0484350	3.425249	3.298435
User_20	Item_1	NA	-0.1175569	0.0484350	3.425249	3.356127
User_22	Item_1	3	0.0191953	0.0484350	3.425249	3.492879
User_24	Item_1	4	0.0747508	0.0484350	3.425249	3.548435
User_25	Item_1	3	-0.1752492	0.0484350	3.425249	3.298435
User_26	Item_1	4	-0.0406338	0.0484350	3.425249	3.433050
User_27	Item_1	3	-0.2434310	0.0484350	3.425249	3.230253
User_28	Item_1	3	-0.0252492	0.0484350	3.425249	3.448435
User_29	Item_1	3	0.0747508	0.0484350	3.425249	3.548435
User_1	Item_2	NA	0.2747508	-0.0919158	3.425249	3.608084
User_3	Item_2	3	0.0292963	-0.0919158	3.425249	3.362630
User_5	Item_2	4	-0.0252492	-0.0919158	3.425249	3.308084
User_6	Item_2	3	0.2414175	-0.0919158	3.425249	3.574751
User_7	Item_2	4	0.0747508	-0.0919158	3.425249	3.408084
User_8	Item_2	NA	0.2414175	-0.0919158	3.425249	3.574751
User_9	Item_2	4	0.3020236	-0.0919158	3.425249	3.635357
User_11	Item_2	3	0.0747508	-0.0919158	3.425249	3.408084
User_12	Item_2	3	-0.3252492	-0.0919158	3.425249	3.008084
User_13	Item_2	NA	-0.3002492	-0.0919158	3.425249	3.033084
User_15	Item_2	3	-0.3141381	-0.0919158	3.425249	3.019195
User_18	Item_2	3	0.0747508	-0.0919158	3.425249	3.408084
User_19	Item_2	3	-0.1752492	-0.0919158	3.425249	3.158084
User_20	Item_2	3	-0.1175569	-0.0919158	3.425249	3.215777
User_21	Item_2	4	-0.0252492	-0.0919158	3.425249	3.308084
User_23	Item_2	NA	0.0191953	-0.0919158	3.425249	3.352529
User_24	Item_2	3	0.0747508	-0.0919158	3.425249	3.408084
User_25	Item_2	4	-0.1752492	-0.0919158	3.425249	3.158084
User_26	Item_2	3	-0.0406338	-0.0919158	3.425249	3.292699
User_27	Item_2	3	-0.2434310	-0.0919158	3.425249	3.089902
User_28	Item_2	4	-0.0252492	-0.0919158	3.425249	3.308084
User_29	Item_2	NA	0.0747508	-0.0919158	3.425249	3.408084
User_30	Item_2	3	0.0191953	-0.0919158	3.425249	3.352529
User_1	Item_3	4	0.2747508	-0.0502492	3.425249	3.649751
User_3	Item_3	3	0.0292963	-0.0502492	3.425249	3.404296

Table 10: Baseline Predictions (Training) *(continued)*

users	item	rating	userBias	itemBias	meanRating	baseline
User_4	Item_3	4	0.0191953	-0.0502492	3.425249	3.394195
User_5	Item_3	3	-0.0252492	-0.0502492	3.425249	3.349751
User_7	Item_3	NA	0.0747508	-0.0502492	3.425249	3.449751
User_9	Item_3	3	0.3020236	-0.0502492	3.425249	3.677024
User_10	Item_3	4	0.0747508	-0.0502492	3.425249	3.449751
User_11	Item_3	3	0.0747508	-0.0502492	3.425249	3.449751
User_12	Item_3	3	-0.3252492	-0.0502492	3.425249	3.049751
User_13	Item_3	3	-0.3002492	-0.0502492	3.425249	3.074751
User_14	Item_3	4	0.3525286	-0.0502492	3.425249	3.727529
User_15	Item_3	3	-0.3141381	-0.0502492	3.425249	3.060862
User_16	Item_3	3	0.0292963	-0.0502492	3.425249	3.404296
User_17	Item_3	3	-0.1752492	-0.0502492	3.425249	3.199751
User_18	Item_3	4	0.0747508	-0.0502492	3.425249	3.449751
User_19	Item_3	3	-0.1752492	-0.0502492	3.425249	3.199751
User_20	Item_3	4	-0.1175569	-0.0502492	3.425249	3.257443
User_21	Item_3	3	-0.0252492	-0.0502492	3.425249	3.349751
User_22	Item_3	3	0.0191953	-0.0502492	3.425249	3.394195
User_23	Item_3	4	0.0191953	-0.0502492	3.425249	3.394195
User_25	Item_3	3	-0.1752492	-0.0502492	3.425249	3.199751
User_26	Item_3	4	-0.0406338	-0.0502492	3.425249	3.334366
User_28	Item_3	3	-0.0252492	-0.0502492	3.425249	3.349751
User_29	Item_3	4	0.0747508	-0.0502492	3.425249	3.449751
User_30	Item_3	3	0.0191953	-0.0502492	3.425249	3.394195
User_1	Item_4	4	0.2747508	-0.0041965	3.425249	3.695803
User_3	Item_4	3	0.0292963	-0.0041965	3.425249	3.450349
User_4	Item_4	3	0.0191953	-0.0041965	3.425249	3.440248
User_5	Item_4	4	-0.0252492	-0.0041965	3.425249	3.395803
User_6	Item_4	NA	0.2414175	-0.0041965	3.425249	3.662470
User_8	Item_4	4	0.2414175	-0.0041965	3.425249	3.662470
User_10	Item_4	4	0.0747508	-0.0041965	3.425249	3.495804
User_11	Item_4	4	0.0747508	-0.0041965	3.425249	3.495804
User_12	Item_4	NA	-0.3252492	-0.0041965	3.425249	3.095804
User_13	Item_4	3	-0.3002492	-0.0041965	3.425249	3.120804
User_14	Item_4	4	0.3525286	-0.0041965	3.425249	3.773581
User_19	Item_4	3	-0.1752492	-0.0041965	3.425249	3.245804
User_20	Item_4	3	-0.1175569	-0.0041965	3.425249	3.303496
User_21	Item_4	3	-0.0252492	-0.0041965	3.425249	3.395803
User_22	Item_4	4	0.0191953	-0.0041965	3.425249	3.440248
User_23	Item_4	3	0.0191953	-0.0041965	3.425249	3.440248
User_25	Item_4	3	-0.1752492	-0.0041965	3.425249	3.245804

Table 10: Baseline Predictions (Training) *(continued)*

users	item	rating	userBias	itemBias	meanRating	baseline
User_26	Item_4	3	-0.0406338	-0.0041965	3.425249	3.380419
User_27	Item_4	3	-0.2434310	-0.0041965	3.425249	3.177622
User_28	Item_4	3	-0.0252492	-0.0041965	3.425249	3.395803
User_30	Item_4	4	0.0191953	-0.0041965	3.425249	3.440248
User_1	Item_5	3	0.2747508	0.0033223	3.425249	3.703322
User_2	Item_5	4	0.0191953	0.0033223	3.425249	3.447767
User_3	Item_5	3	0.0292963	0.0033223	3.425249	3.457868
User_4	Item_5	NA	0.0191953	0.0033223	3.425249	3.447767
User_6	Item_5	4	0.2414175	0.0033223	3.425249	3.669989
User_7	Item_5	3	0.0747508	0.0033223	3.425249	3.503322
User_9	Item_5	4	0.3020236	0.0033223	3.425249	3.730595
User_10	Item_5	3	0.0747508	0.0033223	3.425249	3.503322
User_11	Item_5	3	0.0747508	0.0033223	3.425249	3.503322
User_12	Item_5	4	-0.3252492	0.0033223	3.425249	3.103322
User_15	Item_5	3	-0.3141381	0.0033223	3.425249	3.114433
User_16	Item_5	3	0.0292963	0.0033223	3.425249	3.457868
User_17	Item_5	3	-0.1752492	0.0033223	3.425249	3.253322
User_18	Item_5	3	0.0747508	0.0033223	3.425249	3.503322
User_19	Item_5	4	-0.1752492	0.0033223	3.425249	3.253322
User_20	Item_5	3	-0.1175569	0.0033223	3.425249	3.311015
User_22	Item_5	NA	0.0191953	0.0033223	3.425249	3.447767
User_23	Item_5	4	0.0191953	0.0033223	3.425249	3.447767
User_24	Item_5	4	0.0747508	0.0033223	3.425249	3.503322
User_25	Item_5	3	-0.1752492	0.0033223	3.425249	3.253322
User_26	Item_5	4	-0.0406338	0.0033223	3.425249	3.387938
User_27	Item_5	NA	-0.2434310	0.0033223	3.425249	3.185140
User_28	Item_5	4	-0.0252492	0.0033223	3.425249	3.403322
User_29	Item_5	3	0.0747508	0.0033223	3.425249	3.503322
User_2	Item_6	4	0.0191953	0.0747508	3.425249	3.519195
User_5	Item_6	4	-0.0252492	0.0747508	3.425249	3.474751
User_6	Item_6	4	0.2414175	0.0747508	3.425249	3.741417
User_9	Item_6	4	0.3020236	0.0747508	3.425249	3.802024
User_10	Item_6	3	0.0747508	0.0747508	3.425249	3.574751
User_12	Item_6	3	-0.3252492	0.0747508	3.425249	3.174751
User_13	Item_6	NA	-0.3002492	0.0747508	3.425249	3.199751
User_15	Item_6	4	-0.3141381	0.0747508	3.425249	3.185862
User_16	Item_6	4	0.0292963	0.0747508	3.425249	3.529296
User_17	Item_6	4	-0.1752492	0.0747508	3.425249	3.324751
User_18	Item_6	NA	0.0747508	0.0747508	3.425249	3.574751
User_20	Item_6	4	-0.1175569	0.0747508	3.425249	3.382443

Table 10: Baseline Predictions (Training) *(continued)*

users	item	rating	userBias	itemBias	meanRating	baseline
User_22	Item_6	NA	0.0191953	0.0747508	3.425249	3.519195
User_23	Item_6	3	0.0191953	0.0747508	3.425249	3.519195
User_24	Item_6	3	0.0747508	0.0747508	3.425249	3.574751
User_25	Item_6	3	-0.1752492	0.0747508	3.425249	3.324751
User_26	Item_6	3	-0.0406338	0.0747508	3.425249	3.459366
User_27	Item_6	3	-0.2434310	0.0747508	3.425249	3.256569
User_30	Item_6	3	0.0191953	0.0747508	3.425249	3.519195
User_1	Item_7	3	0.2747508	-0.0568281	3.425249	3.643172
User_3	Item_7	NA	0.0292963	-0.0568281	3.425249	3.397717
User_5	Item_7	3	-0.0252492	-0.0568281	3.425249	3.343172
User_6	Item_7	4	0.2414175	-0.0568281	3.425249	3.609838
User_7	Item_7	4	0.0747508	-0.0568281	3.425249	3.443172
User_8	Item_7	4	0.2414175	-0.0568281	3.425249	3.609838
User_9	Item_7	3	0.3020236	-0.0568281	3.425249	3.670445
User_10	Item_7	3	0.0747508	-0.0568281	3.425249	3.443172
User_11	Item_7	3	0.0747508	-0.0568281	3.425249	3.443172
User_12	Item_7	3	-0.3252492	-0.0568281	3.425249	3.043172
User_13	Item_7	NA	-0.3002492	-0.0568281	3.425249	3.068172
User_14	Item_7	4	0.3525286	-0.0568281	3.425249	3.720950
User_15	Item_7	3	-0.3141381	-0.0568281	3.425249	3.054283
User_18	Item_7	3	0.0747508	-0.0568281	3.425249	3.443172
User_19	Item_7	3	-0.1752492	-0.0568281	3.425249	3.193172
User_20	Item_7	NA	-0.1175569	-0.0568281	3.425249	3.250864
User_21	Item_7	4	-0.0252492	-0.0568281	3.425249	3.343172
User_22	Item_7	3	0.0191953	-0.0568281	3.425249	3.387616
User_23	Item_7	NA	0.0191953	-0.0568281	3.425249	3.387616
User_25	Item_7	3	-0.1752492	-0.0568281	3.425249	3.193172
User_26	Item_7	3	-0.0406338	-0.0568281	3.425249	3.327787
User_29	Item_7	4	0.0747508	-0.0568281	3.425249	3.443172
User_30	Item_7	4	0.0191953	-0.0568281	3.425249	3.387616
User_1	Item_8	4	0.2747508	-0.0252492	3.425249	3.674751
User_2	Item_8	4	0.0191953	-0.0252492	3.425249	3.419195
User_3	Item_8	3	0.0292963	-0.0252492	3.425249	3.429296
User_5	Item_8	NA	-0.0252492	-0.0252492	3.425249	3.374751
User_6	Item_8	3	0.2414175	-0.0252492	3.425249	3.641418
User_7	Item_8	3	0.0747508	-0.0252492	3.425249	3.474751
User_8	Item_8	4	0.2414175	-0.0252492	3.425249	3.641418
User_9	Item_8	4	0.3020236	-0.0252492	3.425249	3.702024
User_11	Item_8	4	0.0747508	-0.0252492	3.425249	3.474751
User_12	Item_8	NA	-0.3252492	-0.0252492	3.425249	3.074751

Table 10: Baseline Predictions (Training) *(continued)*

users	item	rating	userBias	itemBias	meanRating	baseline
User_13	Item_8	3	-0.3002492	-0.0252492	3.425249	3.099751
User_14	Item_8	4	0.3525286	-0.0252492	3.425249	3.752529
User_15	Item_8	3	-0.3141381	-0.0252492	3.425249	3.085862
User_16	Item_8	3	0.0292963	-0.0252492	3.425249	3.429296
User_18	Item_8	4	0.0747508	-0.0252492	3.425249	3.474751
User_19	Item_8	3	-0.1752492	-0.0252492	3.425249	3.224751
User_20	Item_8	3	-0.1175569	-0.0252492	3.425249	3.282443
User_21	Item_8	3	-0.0252492	-0.0252492	3.425249	3.374751
User_22	Item_8	4	0.0191953	-0.0252492	3.425249	3.419195
User_23	Item_8	3	0.0191953	-0.0252492	3.425249	3.419195
User_24	Item_8	3	0.0747508	-0.0252492	3.425249	3.474751
User_25	Item_8	3	-0.1752492	-0.0252492	3.425249	3.224751
User_26	Item_8	4	-0.0406338	-0.0252492	3.425249	3.359366
User_27	Item_8	3	-0.2434310	-0.0252492	3.425249	3.156569
User_28	Item_8	4	-0.0252492	-0.0252492	3.425249	3.374751
User_29	Item_8	3	0.0747508	-0.0252492	3.425249	3.474751
User_30	Item_8	3	0.0191953	-0.0252492	3.425249	3.419195
User_1	Item_9	4	0.2747508	0.0509413	3.425249	3.750941
User_2	Item_9	3	0.0191953	0.0509413	3.425249	3.495386
User_3	Item_9	4	0.0292963	0.0509413	3.425249	3.505487
User_4	Item_9	3	0.0191953	0.0509413	3.425249	3.495386
User_5	Item_9	4	-0.0252492	0.0509413	3.425249	3.450941
User_6	Item_9	3	0.2414175	0.0509413	3.425249	3.717608
User_7	Item_9	3	0.0747508	0.0509413	3.425249	3.550941
User_8	Item_9	3	0.2414175	0.0509413	3.425249	3.717608
User_10	Item_9	4	0.0747508	0.0509413	3.425249	3.550941
User_13	Item_9	4	-0.3002492	0.0509413	3.425249	3.175941
User_14	Item_9	3	0.3525286	0.0509413	3.425249	3.828719
User_17	Item_9	4	-0.1752492	0.0509413	3.425249	3.300941
User_18	Item_9	4	0.0747508	0.0509413	3.425249	3.550941
User_19	Item_9	NA	-0.1752492	0.0509413	3.425249	3.300941
User_20	Item_9	3	-0.1175569	0.0509413	3.425249	3.358634
User_21	Item_9	4	-0.0252492	0.0509413	3.425249	3.450941
User_23	Item_9	3	0.0191953	0.0509413	3.425249	3.495386
User_24	Item_9	4	0.0747508	0.0509413	3.425249	3.550941
User_25	Item_9	3	-0.1752492	0.0509413	3.425249	3.300941
User_26	Item_9	4	-0.0406338	0.0509413	3.425249	3.435557
User_27	Item_9	3	-0.2434310	0.0509413	3.425249	3.232759
User_28	Item_9	3	-0.0252492	0.0509413	3.425249	3.450941
User_2	Item_10	4	0.0191953	-0.0134845	3.425249	3.430960

Table 10: Baseline Predictions (Training) *(continued)*

users	item	rating	userBias	itemBias	meanRating	baseline
User_3	Item_10	4	0.0292963	-0.0134845	3.425249	3.441061
User_4	Item_10	3	0.0191953	-0.0134845	3.425249	3.430960
User_5	Item_10	3	-0.0252492	-0.0134845	3.425249	3.386515
User_6	Item_10	4	0.2414175	-0.0134845	3.425249	3.653182
User_11	Item_10	4	0.0747508	-0.0134845	3.425249	3.486515
User_12	Item_10	3	-0.3252492	-0.0134845	3.425249	3.086515
User_13	Item_10	3	-0.3002492	-0.0134845	3.425249	3.111515
User_14	Item_10	NA	0.3525286	-0.0134845	3.425249	3.764293
User_15	Item_10	NA	-0.3141381	-0.0134845	3.425249	3.097627
User_16	Item_10	3	0.0292963	-0.0134845	3.425249	3.441061
User_17	Item_10	NA	-0.1752492	-0.0134845	3.425249	3.236515
User_19	Item_10	3	-0.1752492	-0.0134845	3.425249	3.236515
User_20	Item_10	4	-0.1175569	-0.0134845	3.425249	3.294208
User_21	Item_10	3	-0.0252492	-0.0134845	3.425249	3.386515
User_22	Item_10	3	0.0191953	-0.0134845	3.425249	3.430960
User_24	Item_10	3	0.0747508	-0.0134845	3.425249	3.486515
User_25	Item_10	4	-0.1752492	-0.0134845	3.425249	3.236515
User_26	Item_10	3	-0.0406338	-0.0134845	3.425249	3.371131
User_27	Item_10	NA	-0.2434310	-0.0134845	3.425249	3.168334
User_29	Item_10	4	0.0747508	-0.0134845	3.425249	3.486515
User_1	Item_11	NA	0.2747508	-0.0363603	3.425249	3.663640
User_2	Item_11	3	0.0191953	-0.0363603	3.425249	3.408084
User_3	Item_11	3	0.0292963	-0.0363603	3.425249	3.418185
User_5	Item_11	3	-0.0252492	-0.0363603	3.425249	3.363640
User_6	Item_11	3	0.2414175	-0.0363603	3.425249	3.630306
User_7	Item_11	4	0.0747508	-0.0363603	3.425249	3.463640
User_9	Item_11	4	0.3020236	-0.0363603	3.425249	3.690912
User_11	Item_11	4	0.0747508	-0.0363603	3.425249	3.463640
User_12	Item_11	NA	-0.3252492	-0.0363603	3.425249	3.063640
User_15	Item_11	3	-0.3141381	-0.0363603	3.425249	3.074751
User_16	Item_11	3	0.0292963	-0.0363603	3.425249	3.418185
User_17	Item_11	3	-0.1752492	-0.0363603	3.425249	3.213640
User_18	Item_11	4	0.0747508	-0.0363603	3.425249	3.463640
User_19	Item_11	3	-0.1752492	-0.0363603	3.425249	3.213640
User_20	Item_11	3	-0.1175569	-0.0363603	3.425249	3.271332
User_22	Item_11	4	0.0191953	-0.0363603	3.425249	3.408084
User_24	Item_11	4	0.0747508	-0.0363603	3.425249	3.463640
User_27	Item_11	3	-0.2434310	-0.0363603	3.425249	3.145458
User_28	Item_11	3	-0.0252492	-0.0363603	3.425249	3.363640
User_29	Item_11	4	0.0747508	-0.0363603	3.425249	3.463640

Table 10: Baseline Predictions (Training) *(continued)*

users	item	rating	userBias	itemBias	meanRating	baseline
User_1	Item_12	4	0.2747508	0.1997508	3.425249	3.899751
User_2	Item_12	3	0.0191953	0.1997508	3.425249	3.644195
User_3	Item_12	4	0.0292963	0.1997508	3.425249	3.654296
User_4	Item_12	4	0.0191953	0.1997508	3.425249	3.644195
User_5	Item_12	3	-0.0252492	0.1997508	3.425249	3.599751
User_7	Item_12	4	0.0747508	0.1997508	3.425249	3.699751
User_9	Item_12	4	0.3020236	0.1997508	3.425249	3.927024
User_10	Item_12	4	0.0747508	0.1997508	3.425249	3.699751
User_11	Item_12	4	0.0747508	0.1997508	3.425249	3.699751
User_12	Item_12	3	-0.3252492	0.1997508	3.425249	3.299751
User_14	Item_12	4	0.3525286	0.1997508	3.425249	3.977529
User_16	Item_12	4	0.0292963	0.1997508	3.425249	3.654296
User_17	Item_12	3	-0.1752492	0.1997508	3.425249	3.449751
User_18	Item_12	4	0.0747508	0.1997508	3.425249	3.699751
User_19	Item_12	3	-0.1752492	0.1997508	3.425249	3.449751
User_20	Item_12	3	-0.1175569	0.1997508	3.425249	3.507443
User_21	Item_12	3	-0.0252492	0.1997508	3.425249	3.599751
User_22	Item_12	3	0.0191953	0.1997508	3.425249	3.644195
User_23	Item_12	4	0.0191953	0.1997508	3.425249	3.644195
User_24	Item_12	4	0.0747508	0.1997508	3.425249	3.699751
User_26	Item_12	3	-0.0406338	0.1997508	3.425249	3.584366
User_27	Item_12	4	-0.2434310	0.1997508	3.425249	3.381569
User_28	Item_12	NA	-0.0252492	0.1997508	3.425249	3.599751
User_29	Item_12	4	0.0747508	0.1997508	3.425249	3.699751
User_30	Item_12	4	0.0191953	0.1997508	3.425249	3.644195
User_1	Item_13	3	0.2747508	-0.0339448	3.425249	3.666055
User_2	Item_13	3	0.0191953	-0.0339448	3.425249	3.410500
User_3	Item_13	4	0.0292963	-0.0339448	3.425249	3.420601
User_4	Item_13	3	0.0191953	-0.0339448	3.425249	3.410500
User_6	Item_13	4	0.2414175	-0.0339448	3.425249	3.632722
User_7	Item_13	3	0.0747508	-0.0339448	3.425249	3.466055
User_8	Item_13	3	0.2414175	-0.0339448	3.425249	3.632722
User_10	Item_13	3	0.0747508	-0.0339448	3.425249	3.466055
User_11	Item_13	3	0.0747508	-0.0339448	3.425249	3.466055
User_12	Item_13	3	-0.3252492	-0.0339448	3.425249	3.066055
User_13	Item_13	NA	-0.3002492	-0.0339448	3.425249	3.091055
User_14	Item_13	4	0.3525286	-0.0339448	3.425249	3.743833
User_16	Item_13	4	0.0292963	-0.0339448	3.425249	3.420601
User_17	Item_13	3	-0.1752492	-0.0339448	3.425249	3.216055
User_18	Item_13	3	0.0747508	-0.0339448	3.425249	3.466055

Table 10: Baseline Predictions (Training) *(continued)*

users	item	rating	userBias	itemBias	meanRating	baseline
User_19	Item_13	3	-0.1752492	-0.0339448	3.425249	3.216055
User_20	Item_13	4	-0.1175569	-0.0339448	3.425249	3.273747
User_21	Item_13	4	-0.0252492	-0.0339448	3.425249	3.366055
User_23	Item_13	4	0.0191953	-0.0339448	3.425249	3.410500
User_24	Item_13	3	0.0747508	-0.0339448	3.425249	3.466055
User_25	Item_13	3	-0.1752492	-0.0339448	3.425249	3.216055
User_26	Item_13	3	-0.0406338	-0.0339448	3.425249	3.350671
User_27	Item_13	4	-0.2434310	-0.0339448	3.425249	3.147873
User_28	Item_13	4	-0.0252492	-0.0339448	3.425249	3.366055
User_29	Item_13	NA	0.0747508	-0.0339448	3.425249	3.466055
User_1	Item_14	NA	0.2747508	-0.1311315	3.425249	3.568869
User_2	Item_14	3	0.0191953	-0.1311315	3.425249	3.313313
User_4	Item_14	4	0.0191953	-0.1311315	3.425249	3.313313
User_6	Item_14	4	0.2414175	-0.1311315	3.425249	3.535535
User_7	Item_14	4	0.0747508	-0.1311315	3.425249	3.368869
User_8	Item_14	4	0.2414175	-0.1311315	3.425249	3.535535
User_9	Item_14	3	0.3020236	-0.1311315	3.425249	3.596141
User_12	Item_14	3	-0.3252492	-0.1311315	3.425249	2.968869
User_13	Item_14	3	-0.3002492	-0.1311315	3.425249	2.993869
User_15	Item_14	3	-0.3141381	-0.1311315	3.425249	2.979980
User_16	Item_14	3	0.0292963	-0.1311315	3.425249	3.323414
User_20	Item_14	3	-0.1175569	-0.1311315	3.425249	3.176561
User_22	Item_14	4	0.0191953	-0.1311315	3.425249	3.313313
User_23	Item_14	3	0.0191953	-0.1311315	3.425249	3.313313
User_24	Item_14	NA	0.0747508	-0.1311315	3.425249	3.368869
User_26	Item_14	3	-0.0406338	-0.1311315	3.425249	3.253484
User_27	Item_14	3	-0.2434310	-0.1311315	3.425249	3.050687
User_28	Item_14	NA	-0.0252492	-0.1311315	3.425249	3.268868
User_29	Item_14	3	0.0747508	-0.1311315	3.425249	3.368869
User_30	Item_14	3	0.0191953	-0.1311315	3.425249	3.313313
User_1	Item_15	4	0.2747508	0.0247508	3.425249	3.724751
User_3	Item_15	4	0.0292963	0.0247508	3.425249	3.479296
User_4	Item_15	4	0.0191953	0.0247508	3.425249	3.469195
User_5	Item_15	3	-0.0252492	0.0247508	3.425249	3.424751
User_6	Item_15	4	0.2414175	0.0247508	3.425249	3.691418
User_9	Item_15	4	0.3020236	0.0247508	3.425249	3.752024
User_11	Item_15	3	0.0747508	0.0247508	3.425249	3.524751
User_13	Item_15	3	-0.3002492	0.0247508	3.425249	3.149751
User_14	Item_15	3	0.3525286	0.0247508	3.425249	3.802529
User_15	Item_15	3	-0.3141381	0.0247508	3.425249	3.135862

Table 10: Baseline Predictions (Training) *(continued)*

users	item	rating	userBias	itemBias	meanRating	baseline
User_16	Item_15	4	0.0292963	0.0247508	3.425249	3.479296
User_17	Item_15	NA	-0.1752492	0.0247508	3.425249	3.274751
User_18	Item_15	3	0.0747508	0.0247508	3.425249	3.524751
User_19	Item_15	4	-0.1752492	0.0247508	3.425249	3.274751
User_20	Item_15	3	-0.1175569	0.0247508	3.425249	3.332443
User_21	Item_15	3	-0.0252492	0.0247508	3.425249	3.424751
User_22	Item_15	NA	0.0191953	0.0247508	3.425249	3.469195
User_23	Item_15	NA	0.0191953	0.0247508	3.425249	3.469195
User_24	Item_15	NA	0.0747508	0.0247508	3.425249	3.524751
User_25	Item_15	4	-0.1752492	0.0247508	3.425249	3.274751
User_27	Item_15	3	-0.2434310	0.0247508	3.425249	3.206569
User_28	Item_15	3	-0.0252492	0.0247508	3.425249	3.424751
User_29	Item_15	3	0.0747508	0.0247508	3.425249	3.524751
User_30	Item_15	4	0.0191953	0.0247508	3.425249	3.469195

Table 11: Baseline Predictions (Testing)

users	item	rating	userBias	itemBias	meanRating	baseline
User_3	Item_1	2	0.0292963	0.0484350	3.425249	3.502981
User_5	Item_1	5	-0.0252492	0.0484350	3.425249	3.448435
User_8	Item_1	2	0.2414175	0.0484350	3.425249	3.715102
User_10	Item_1	2	0.0747508	0.0484350	3.425249	3.548435
User_15	Item_1	5	-0.3141381	0.0484350	3.425249	3.159546
User_18	Item_1	2	0.0747508	0.0484350	3.425249	3.548435
User_21	Item_1	2	-0.0252492	0.0484350	3.425249	3.448435
User_23	Item_1	2	0.0191953	0.0484350	3.425249	3.492879
User_30	Item_1	5	0.0191953	0.0484350	3.425249	3.492879
User_2	Item_2	1	0.0191953	-0.0919158	3.425249	3.352529
User_4	Item_2	1	0.0191953	-0.0919158	3.425249	3.352529
User_10	Item_2	5	0.0747508	-0.0919158	3.425249	3.408084
User_14	Item_2	5	0.3525286	-0.0919158	3.425249	3.685862
User_16	Item_2	2	0.0292963	-0.0919158	3.425249	3.362630
User_17	Item_2	5	-0.1752492	-0.0919158	3.425249	3.158084
User_22	Item_2	1	0.0191953	-0.0919158	3.425249	3.352529
User_2	Item_3	2	0.0191953	-0.0502492	3.425249	3.394195
User_6	Item_3	5	0.2414175	-0.0502492	3.425249	3.616417
User_8	Item_3	2	0.2414175	-0.0502492	3.425249	3.616417
User_24	Item_3	5	0.0747508	-0.0502492	3.425249	3.449751
User_27	Item_3	2	-0.2434310	-0.0502492	3.425249	3.131569

Table 11: Baseline Predictions (Testing) *(continued)*

users	item	rating	userBias	itemBias	meanRating	baseline
User_2	Item_4	5	0.0191953	-0.0041965	3.425249	3.440248
User_7	Item_4	1	0.0747508	-0.0041965	3.425249	3.495804
User_9	Item_4	5	0.3020236	-0.0041965	3.425249	3.723076
User_15	Item_4	2	-0.3141381	-0.0041965	3.425249	3.106915
User_16	Item_4	1	0.0292963	-0.0041965	3.425249	3.450349
User_17	Item_4	2	-0.1752492	-0.0041965	3.425249	3.245804
User_18	Item_4	5	0.0747508	-0.0041965	3.425249	3.495804
User_24	Item_4	2	0.0747508	-0.0041965	3.425249	3.495804
User_29	Item_4	2	0.0747508	-0.0041965	3.425249	3.495804
User_5	Item_5	5	-0.0252492	0.0033223	3.425249	3.403322
User_8	Item_5	2	0.2414175	0.0033223	3.425249	3.669989
User_13	Item_5	5	-0.3002492	0.0033223	3.425249	3.128322
User_14	Item_5	1	0.3525286	0.0033223	3.425249	3.781100
User_21	Item_5	5	-0.0252492	0.0033223	3.425249	3.403322
User_30	Item_5	1	0.0191953	0.0033223	3.425249	3.447767
User_1	Item_6	5	0.2747508	0.0747508	3.425249	3.774751
User_3	Item_6	5	0.0292963	0.0747508	3.425249	3.529296
User_4	Item_6	1	0.0191953	0.0747508	3.425249	3.519195
User_7	Item_6	1	0.0747508	0.0747508	3.425249	3.574751
User_8	Item_6	5	0.2414175	0.0747508	3.425249	3.741417
User_11	Item_6	5	0.0747508	0.0747508	3.425249	3.574751
User_14	Item_6	5	0.3525286	0.0747508	3.425249	3.852529
User_19	Item_6	2	-0.1752492	0.0747508	3.425249	3.324751
User_21	Item_6	5	-0.0252492	0.0747508	3.425249	3.474751
User_28	Item_6	2	-0.0252492	0.0747508	3.425249	3.474751
User_29	Item_6	1	0.0747508	0.0747508	3.425249	3.574751
User_2	Item_7	5	0.0191953	-0.0568281	3.425249	3.387616
User_4	Item_7	2	0.0191953	-0.0568281	3.425249	3.387616
User_16	Item_7	5	0.0292963	-0.0568281	3.425249	3.397717
User_17	Item_7	2	-0.1752492	-0.0568281	3.425249	3.193172
User_24	Item_7	5	0.0747508	-0.0568281	3.425249	3.443172
User_27	Item_7	2	-0.2434310	-0.0568281	3.425249	3.124990
User_28	Item_7	5	-0.0252492	-0.0568281	3.425249	3.343172
User_4	Item_8	5	0.0191953	-0.0252492	3.425249	3.419195
User_10	Item_8	5	0.0747508	-0.0252492	3.425249	3.474751
User_17	Item_8	1	-0.1752492	-0.0252492	3.425249	3.224751
User_9	Item_9	5	0.3020236	0.0509413	3.425249	3.778214
User_11	Item_9	1	0.0747508	0.0509413	3.425249	3.550941
User_12	Item_9	5	-0.3252492	0.0509413	3.425249	3.150941
User_15	Item_9	2	-0.3141381	0.0509413	3.425249	3.162052

Table 11: Baseline Predictions (Testing) *(continued)*

users	item	rating	userBias	itemBias	meanRating	baseline
User_16	Item_9	1	0.0292963	0.0509413	3.425249	3.505487
User_22	Item_9	2	0.0191953	0.0509413	3.425249	3.495386
User_29	Item_9	5	0.0747508	0.0509413	3.425249	3.550941
User_30	Item_9	2	0.0191953	0.0509413	3.425249	3.495386
User_1	Item_10	2	0.2747508	-0.0134845	3.425249	3.686516
User_7	Item_10	5	0.0747508	-0.0134845	3.425249	3.486515
User_8	Item_10	2	0.2414175	-0.0134845	3.425249	3.653182
User_9	Item_10	5	0.3020236	-0.0134845	3.425249	3.713788
User_10	Item_10	2	0.0747508	-0.0134845	3.425249	3.486515
User_18	Item_10	1	0.0747508	-0.0134845	3.425249	3.486515
User_23	Item_10	2	0.0191953	-0.0134845	3.425249	3.430960
User_28	Item_10	1	-0.0252492	-0.0134845	3.425249	3.386515
User_30	Item_10	1	0.0191953	-0.0134845	3.425249	3.430960
User_4	Item_11	1	0.0191953	-0.0363603	3.425249	3.408084
User_8	Item_11	2	0.2414175	-0.0363603	3.425249	3.630306
User_10	Item_11	1	0.0747508	-0.0363603	3.425249	3.463640
User_13	Item_11	1	-0.3002492	-0.0363603	3.425249	3.088640
User_14	Item_11	5	0.3525286	-0.0363603	3.425249	3.741417
User_21	Item_11	5	-0.0252492	-0.0363603	3.425249	3.363640
User_23	Item_11	5	0.0191953	-0.0363603	3.425249	3.408084
User_25	Item_11	2	-0.1752492	-0.0363603	3.425249	3.213640
User_26	Item_11	1	-0.0406338	-0.0363603	3.425249	3.348255
User_30	Item_11	1	0.0191953	-0.0363603	3.425249	3.408084
User_6	Item_12	5	0.2414175	0.1997508	3.425249	3.866417
User_8	Item_12	5	0.2414175	0.1997508	3.425249	3.866417
User_13	Item_12	5	-0.3002492	0.1997508	3.425249	3.324751
User_15	Item_12	5	-0.3141381	0.1997508	3.425249	3.310862
User_25	Item_12	2	-0.1752492	0.1997508	3.425249	3.449751
User_5	Item_13	2	-0.0252492	-0.0339448	3.425249	3.366055
User_9	Item_13	1	0.3020236	-0.0339448	3.425249	3.693328
User_15	Item_13	1	-0.3141381	-0.0339448	3.425249	3.077166
User_22	Item_13	2	0.0191953	-0.0339448	3.425249	3.410500
User_30	Item_13	1	0.0191953	-0.0339448	3.425249	3.410500
User_3	Item_14	5	0.0292963	-0.1311315	3.425249	3.323414
User_5	Item_14	1	-0.0252492	-0.1311315	3.425249	3.268868
User_10	Item_14	5	0.0747508	-0.1311315	3.425249	3.368869
User_11	Item_14	2	0.0747508	-0.1311315	3.425249	3.368869
User_14	Item_14	5	0.3525286	-0.1311315	3.425249	3.646646
User_17	Item_14	2	-0.1752492	-0.1311315	3.425249	3.118869
User_18	Item_14	2	0.0747508	-0.1311315	3.425249	3.368869

Table 11: Baseline Predictions (Testing) *(continued)*

users	item	rating	userBias	itemBias	meanRating	baseline
User_19	Item_14	2	-0.1752492	-0.1311315	3.425249	3.118869
User_21	Item_14	2	-0.0252492	-0.1311315	3.425249	3.268868
User_25	Item_14	5	-0.1752492	-0.1311315	3.425249	3.118869
User_2	Item_15	5	0.0191953	0.0247508	3.425249	3.469195
User_7	Item_15	1	0.0747508	0.0247508	3.425249	3.524751
User_8	Item_15	2	0.2414175	0.0247508	3.425249	3.691418
User_10	Item_15	2	0.0747508	0.0247508	3.425249	3.524751
User_12	Item_15	2	-0.3252492	0.0247508	3.425249	3.124751
User_26	Item_15	5	-0.0406338	0.0247508	3.425249	3.409366

Calculate the RMSE for the baseline predictors for both your training data and your test data.

We compute the root mean squared error (RMSE) for both the training and testing data:

```
# compute baseline RMSE for training set
rmseTrainBaseline<-RMSE_baseline(train_table$rating,train_table$baseline)
# compute baseline RMSE for testing set
rmseTestBaseline<-RMSE_baseline(test_table$rating,test_table$baseline)
```

Summarize your results.

We create a summary table to compare the root mean squared error (RMSE) of the (raw) mean rating and the baseline predictions for both the training and testing data sets:

```
# create summary table
rmses<-matrix(c(rmseTrain,rmseTrainBaseline,rmseTest,rmseTestBaseline),
  byrow=T,nrow=2)
setLabels<-c('Train','Test')
rmseTypeLabels<-c('Mean (Raw)','Baseline')
colnames(rmses)<-rmseTypeLabels
rownames(rmses)<-setLabels
```

Table 12: Toy Data Set RMSE Summary

	Mean (Raw)	Baseline
Train	0.4943807	0.4581305
Test	1.7665942	1.7517292

From the above table we see that baseline approach improves upon the raw average (mean) approach for both the training and testing data sets.

Software

This project was created using base R [R Core Team, 2019] and the R markdown [Allaire et al., 2018], tint [Eddelbuettel and Gilligan, 2019], kableExtra [Zhu, 2019], and tidyverse [Wickham, 2017] libraries.

References

JJ Allaire, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, and Winston Chang. *rmarkdown: Dynamic Documents for R*, 2018. URL https://CRAN.R-project.org/package=rmarkdown. R package version 1.10.

Dirk Eddelbuettel and Jonathan Gilligan. *tint: 'tint' is not 'Tufte'*, 2019. URL https://CRAN.R-project.org/package=tint. R package version 0.1.1.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019. URL https://www.R-project.org/.

Hadley Wickham. *tidyverse: Easily Install and Load the 'Tidyverse'*, 2017. URL https://CRAN.R-project.org/package=tidyverse. R package version 1.2.1.

Hao Zhu. *kableExtra: Construct Complex Table with 'kable' and Pipe Syntax*, 2019. URL https://CRAN.R-project.org/package=kableExtra. R package version 1.1.0.