HW1 Part2 Flux

Zoey Li

September 10, 2017

1 Submit job to flux

1.1 write a batch script

Below is my flux batch script `HW1_lizeyu.pbs`

```
####  PBS preamble
#PBS -N R_HW1_lizeyu
#PBS -m abe
#PBS -j oe

#PBS -l nodes=1:ppn=5,mem=1gb,walltime=15:00
#PBS -V

#PBS -A stats700f17_flux
#PBS -q flux
####  End PBS preamble

#  Show list of CPUs you ran on, if you're running under PBS
if [ -n "$PBS_NODEFILE" ]; then cat $PBS_NODEFILE; fi

#  Change to the directory you submitted from
if [ -n "$PBS_O_WORKDIR" ]; then cd $PBS_O_WORKDIR; fi
pwd

#  Put your job commands after this line
R CMD BATCH --vanilla HW1_lizeyu.R
```

1.2 Prepare R script to submit

Below is my R script that will save number outputs to a text file called `HW1_lizeyu_number.txt` and save image outputs to a pdf file called `HW1_lizeyu_images.pdf`.

```r
# HW1_lizeyu.R
suppressMessages(library(ggplot2))
suppressMessages(library(gridExtra))
suppressMessages(library(reshape2))
suppressMessages(library(ggbiplot))
suppressMessages(library(corrplot))
load("NBA_data.rda")
dat <- data

# save numbers to text file
sink(file="HW1_lizeyu_numbers.txt")
print("Dimension of the dataset is")
dim(dat)
```

```r
print("Five-number summary of each variables")
summary(dat)

print("Pairwise correlations")
M = cor(dat)
M
sink()

#######################################
## save images to pdfs
pdf(file = "./HW1_lizeyu_images.pdf")
suppressMessages(library(ggplot2))
suppressMessages(library(gridExtra))
suppressMessages(library(reshape2))
suppressMessages(library(ggbiplot))
suppressMessages(library(corrplot))
load("NBA_data.rda")
dat <- data

# Histograms of proportion variables:

hist2 <-
  ggplot(data, aes(x=PP2)) +
  geom_histogram(aes(y=..density..),
                 binwidth = 0.05,
                 colour="black", fill="white") +
  geom_density(alpha=.2) +
  xlab("2-point field goal percentage") +
  scale_x_continuous(limits = c(0,1))+
  ggtitle("Distribution of PP2 \n (2-point field goals percentage)")

hist3 <-
  ggplot(dat, aes(x=PP3)) +
  geom_histogram(aes(y=..density..),
                 binwidth = 0.05,
                 colour="black", fill="white") +
  geom_density(alpha=.2) +
  xlab("3-point field goal percentage") +
  scale_x_continuous(limits = c(0,1))+
  ggtitle("Distribution of PP3 \n (3-point field goals percentage)")

grid.arrange(hist2, hist3, ncol = 2)

## Histograms of variables with values large in magnitude:

g_hist <-
  ggplot(dat, aes(x=G)) +
  geom_histogram(aes(y=..density..),
                 binwidth=2,
                 colour="black", fill="white") +
  geom_density(alpha=.2) +
  xlab("Number of Games") +
  ggtitle("Distribution of Games Played (G)")
```

```r
mp_hist <-
  ggplot(dat, aes(x=MP)) +
  geom_histogram(aes(y=..density..),
                 binwidth=2,
                 colour="black", fill="white") +
  geom_density(alpha=.2) +
  xlab("Minutes Played Per Game") +
  ggtitle("Distribution of Minutes Played Per Game (MP)")

pts_hist <-
  ggplot(dat, aes(x=MP)) +
  geom_histogram(aes(y=..density..),
                 binwidth=2,
                 colour="black", fill="white") +
  geom_density(alpha=.2) +
  xlab("Number of Points Scored Per Game") +
  ggtitle("Distribution of Number of Points Scored Per Game (PTS)")

grid.arrange(mp_hist, g_hist, pts_hist, nrow = 3)

## Boxplots of rest of the continuous varialbes:

suppressMessages(boxplt <-
                   ggplot(dat = melt(dat[,-c(1,2,7,10,19)]), aes(x=variable, y=value)) +
                   geom_boxplot(aes(fill=variable), show.legend = FALSE) +
                   ggtitle("Boxplots of continuous data variables") +
                   theme(axis.text.x = element_text(angle=45, vjust=1, hjust=1))
)
boxplt

# k-means clustering

print("Choosing the number of clusters")

dat_norm = scale(dat)
wss <- (nrow(dat_norm)-1)*sum(apply(dat_norm,2,var))
for (i in 2:15) wss[i] <- kmeans(dat_norm,centers=i)$betweenss/kmeans(dat_norm,centers=i)$totss
df = data.frame(variance = wss[-1])

kpp_elbow_plot <- ggplot(df, aes(x = 1:14, y = variance))  +
  geom_line(size=1) + labs(x = "model", y = "ss") + geom_point(size=3) +
  xlab("Number of Clusters") + ylab("Percentage of Variance explained (%)")
kpp_elbow_plot

set.seed(23462)
clust_kmeans = kmeans(dat_norm, 4, iter.max=100, algorithm='Lloyd')


# Pairwise correlations
M = cor(dat)
corrplot(M, method="number", title = "\n Correlation Heat in Numbers")

# Additional exploratorary data analysis (PCA)
```

```r
pca <- princomp(dat[,-1], cor=TRUE)
biplt <- ggbiplot(pca, alpha = 0.6, title="biplot")
biplt

scree <- qplot(1:length(pca$sdev), pca$sdev^2/sum(pca$sdev^2), geom='line',
               ylab='Component Variance', xlab='Component')
scree


## Visualize clusters on PCA biplot
kmeans_result_4= as.factor(clust_kmeans$cluster)

plot_with_kmeans = function(data1_norm, kmeans_result_4){
  pca = princomp(dat_norm, cor=TRUE)
  pca_temp = data.frame(pca$scores[,1],pca$scores[,2])
  pca_temp = cbind(pca_temp, kmeans_result_4)
  names(pca_temp) = c('PC1', 'PC2', 'k-means (4 clusters)')

  pca_tempML = melt(pca_temp, id.vars=c('PC1', 'PC2'))

  plot = ggplot(pca_tempML, aes(x=PC1, y=PC2, color=value)) +
    geom_point() + facet_grid(~variable) + theme(legend.position='bottom')
  plot
}
plot_with_kmeans(dat_norm, kmeans_result_4)

dev.off()
```

1.3 Submitting job on flux command line

```
$ cd /home/lizeyu/Hw1_lizeyu
$ module load R
$ qsub HW1_lizeyu.pbs
$ qstat -u lizeyu
$ vim HW1_lizeyu.Rout
```

1.4 Outputs

1. `HW1_lizeyu.Rout` (results of runtime included at the bottom)

```
> proc.time()
   user  system elapsed
  2.249   0.088   5.616
```

2. `HW1_lizeyu_numbers.txt`
3. `HW1_lizeyu_images.pdf`

2 Parallel computing

2.1 Write a batch script

Below is my flux batch script for job arrays `HW1_lizeyu_parallel.pbs`

```
####  PBS preamble
#PBS -N R_HW1_lizeyu_parallel5
#PBS -t 1-3
#PBS -l nodes=1:ppn=5,mem=1gb,walltime=15:00
#PBS -V

#PBS -A stats700f17_flux
#PBS -q flux
####  End PBS preamble

# Showlist of CPUs you ran on, if you're running under PBS
if [ -n "$PBS_NODEFILE" ]; then cat $PBS_NODEFILE; fi

#  Change to the directory you submitted from
if [ -n "$PBS_O_WORKDIR" ]; then cd $PBS_O_WORKDIR; fi
pwd

R CMD BATCH --vanilla HW1_lizeyu_parallel_part${PBS_ARRAYID}.R
```

2.2 Next, I cut my `HW1_lizeyu.R` into three independent parts.

1. `HW1_lizeyu_parallel_part1` contains summary statistics(numbers).

```r
suppressMessages(library(ggplot2))
suppressMessages(library(gridExtra))
suppressMessages(library(reshape2))
suppressMessages(library(ggbiplot))
suppressMessages(library(corrplot))

load("NBA_data.rda")
dat <- data

sink(file="HW1_lizeyu_parallel_part1.txt")
print("Dimension of the dataset is")
dim(dat)

print("Five-number summary of each variables")
summary(dat)

print("Pairwise correlations")
M = cor(dat)
M
sink()
```

2. `HW1_lizeyu_parallel_part2` contains summary plots (histograms and boxplots).

```r
suppressMessages(library(ggplot2))
suppressMessages(library(gridExtra))
```

```r
suppressMessages(library(reshape2))
suppressMessages(library(ggbiplot))
suppressMessages(library(corrplot))

load("NBA_data.rda")
dat <- data

pdf(file = "./HW1_lizeyu_parallel_part2.pdf")
# Histograms of proportion variables:

hist2 <-
  ggplot(data, aes(x=PP2)) +
  geom_histogram(aes(y=..density..),
                 binwidth = 0.05,
                 colour="black", fill="white") +
  geom_density(alpha=.2) +
  xlab("2-point field goal percentage") +
  scale_x_continuous(limits = c(0,1))+
  ggtitle("Distribution of PP2 \n (2-point field goals percentage)")

hist3 <-
  ggplot(dat, aes(x=PP3)) +
  geom_histogram(aes(y=..density..),
                 binwidth = 0.05,
                 colour="black", fill="white") +
  geom_density(alpha=.2) +
  xlab("3-point field goal percentage") +
  scale_x_continuous(limits = c(0,1))+
  ggtitle("Distribution of PP3 \n (3-point field goals percentage)")

grid.arrange(hist2, hist3, ncol = 2)

## Histograms of variables with values large in magnitude:

g_hist <-
  ggplot(dat, aes(x=G)) +
  geom_histogram(aes(y=..density..),
                 binwidth=2,
                 colour="black", fill="white") +
  geom_density(alpha=.2) +
  xlab("Number of Games") +
  ggtitle("Distribution of Games Played (G)")

mp_hist <-
  ggplot(dat, aes(x=MP)) +
  geom_histogram(aes(y=..density..),
                 binwidth=2,
                 colour="black", fill="white") +
  geom_density(alpha=.2) +
  xlab("Minutes Played Per Game") +
  ggtitle("Distribution of Minutes Played Per Game (MP)")

pts_hist <-
```

```r
  ggplot(dat, aes(x=MP)) +
  geom_histogram(aes(y=..density..),
                 binwidth=2,
                 colour="black", fill="white") +
  geom_density(alpha=.2) +
  xlab("Number of Points Scored Per Game") +
  ggtitle("Distribution of Number of Points Scored Per Game (PTS)")

grid.arrange(mp_hist, g_hist, pts_hist, nrow = 3)

## Boxplots of rest of the continuous varialbes:

suppressMessages(boxplt <-
                   ggplot(dat = melt(dat[,-c(1,2,7,10,19)]), aes(x=variable, y=value)) +
                   geom_boxplot(aes(fill=variable), show.legend = FALSE) +
                   ggtitle("Boxplots of continuous data variables") +
                   theme(axis.text.x = element_text(angle=45, vjust=1, hjust=1))
)
boxplt

# k-means clustering

print("Choosing the number of clusters")

dat_norm = scale(dat)
wss <- (nrow(dat_norm)-1)*sum(apply(dat_norm,2,var))
for (i in 2:15) wss[i] <- kmeans(dat_norm,centers=i)$betweenss/kmeans(dat_norm,centers=i)$totss
df = data.frame(variance = wss[-1])

kpp_elbow_plot <- ggplot(df, aes(x = 1:14, y = variance))  +
  geom_line(size=1) + labs(x = "model", y = "ss") + geom_point(size=3) +
  xlab("Number of Clusters") + ylab("Percentage of Variance explained (%)")
kpp_elbow_plot


set.seed(23462)
clust_kmeans = kmeans(dat_norm, 4, iter.max=100, algorithm='Lloyd')


# Pairwise correlations
M = cor(dat)
corrplot(M, method="number", title = "\n Correlation Heat in Numbers")

# Additional exploratorary data analysis (PCA)

pca <- princomp(dat[,-1], cor=TRUE)
biplt <- ggbiplot(pca, alpha = 0.6, title="biplot")
biplt

scree <- qplot(1:length(pca$sdev), pca$sdev^2/sum(pca$sdev^2), geom='line',
               ylab='Component Variance', xlab='Component')
scree
```

```r
## Visualize clusters on PCA biplot
kmeans_result_4= as.factor(clust_kmeans$cluster)

plot_with_kmeans = function(data1_norm, kmeans_result_4){
  pca = princomp(dat_norm, cor=TRUE)
  pca_temp = data.frame(pca$scores[,1],pca$scores[,2])
  pca_temp = cbind(pca_temp, kmeans_result_4)
  names(pca_temp) = c('PC1', 'PC2', 'k-means (4 clusters)')

  pca_tempML = melt(pca_temp, id.vars=c('PC1', 'PC2'))

  plot = ggplot(pca_tempML, aes(x=PC1, y=PC2, color=value)) +
    geom_point() + facet_grid(~variable) + theme(legend.position='bottom')
  plot
}
plot_with_kmeans(dat_norm, kmeans_result_4)

dev.off()
```

3. `HW1_lizeyu_parallel_part3` contains k-means analysis plots.

```r
suppressMessages(library(ggplot2))
suppressMessages(library(gridExtra))
suppressMessages(library(reshape2))
suppressMessages(library(ggbiplot))
suppressMessages(library(corrplot))

load("NBA_data.rda")
dat <- data

pdf(file = "./HW1_lizeyu_parallel_part3.pdf")
# k-means clustering
print("Choosing the number of clusters")

dat_norm = scale(dat)
wss <- (nrow(dat_norm)-1)*sum(apply(dat_norm,2,var))
for (i in 2:15) wss[i] <- kmeans(dat_norm,centers=i)$betweenss/kmeans(dat_norm,centers=i)$totss
df = data.frame(variance = wss[-1])

kpp_elbow_plot <- ggplot(df, aes(x = 1:14, y = variance))  +
  geom_line(size=1) + labs(x = "model", y = "ss") + geom_point(size=3) +
  xlab("Number of Clusters") + ylab("Percentage of Variance explained (%)")
kpp_elbow_plot

set.seed(23462)
clust_kmeans = kmeans(dat_norm, 4, iter.max=100, algorithm='Lloyd')


# Pairwise correlations
M = cor(dat)
corrplot(M, method="number", title = "\n Correlation Heat in Numbers")

# Additional exploratory data analysis (PCA)
pca <- princomp(dat[,-1], cor=TRUE)
```

```
biplt <- ggbiplot(pca, alpha = 0.6, title="biplot")
biplt

scree <- qplot(1:length(pca$sdev), pca$sdev^2/sum(pca$sdev^2), geom='line',
               ylab='Component Variance', xlab='Component')
scree


## Visualize clusters on PCA biplot
kmeans_result_4= as.factor(clust_kmeans$cluster)

plot_with_kmeans = function(data1_norm, kmeans_result_4){
  pca = princomp(dat_norm, cor=TRUE)
  pca_temp = data.frame(pca$scores[,1],pca$scores[,2])
  pca_temp = cbind(pca_temp, kmeans_result_4)
  names(pca_temp) = c('PC1', 'PC2', 'k-means (4 clusters)')

  pca_tempML = melt(pca_temp, id.vars=c('PC1', 'PC2'))

  plot = ggplot(pca_tempML, aes(x=PC1, y=PC2, color=value)) +
    geom_point() + facet_grid(~variable) + theme(legend.position='bottom')
  plot
}
plot_with_kmeans(dat_norm, kmeans_result_4)

dev.off()
```

2.3 Submitting job on flux command line

```
$ cd /home/lizeyu/Hw1_lizeyu
$ module load R
$ qsub HW1_lizeyu_parallel.pbs
$ qstat -u lizeyu
```

2.4 Outputs

1. Summary Statistics Script Runtime (`HW1_lizeyu_parallel_part1.Rout`)

```
> proc.time()
   user  system elapsed
  0.389   0.052   1.011
```

2. Summary Plots Script Runtime (`HW1_lizeyu_parallel_part2.Rout`)

```
> proc.time()
   user  system elapsed
  2.233   0.068   3.067
```

3. K-means Clustering Analysis Script Runtime (`HW1_lizeyu_parallel_part3.Rout`)

```
> proc.time()
   user  system elapsed
  1.214   0.061   1.986
```

4. Summary statistics pdf `HW1_lizeyu_parallel_part1.txt`

5. Summary plots pdf `HW1_lizeyu_parallel_part2.pdf`
6. k-means clustering pdf `HW1_lizeyu_parallel_part3.pdf`