Assignment 3: Hadoop, MapReduce, R, and Mahout Part 2

Justin Sallese

April 22, 2017

Below is the data analysis of the data found in assignment 2 using hive in cloudera and the R programming language to generate graphs. As the specific data set to be used was not specified, the data found in assignment 2 shall be used here.The R code that is found here is based off of the code that is found in assignment 2. But the data was sorted in hive in cloudera.

The analysis that is to be done is to find out what is happening on safari and firefox when you access you favourite website, which in this case is gmail. The data shall be sorted using hive in cloudera and then there will be graphs made in the R programming language.

Below is a link to the creation of the tables within hive that help sort the data for analysis.

Safari Data Table Creation

Firefox Data Table Creation

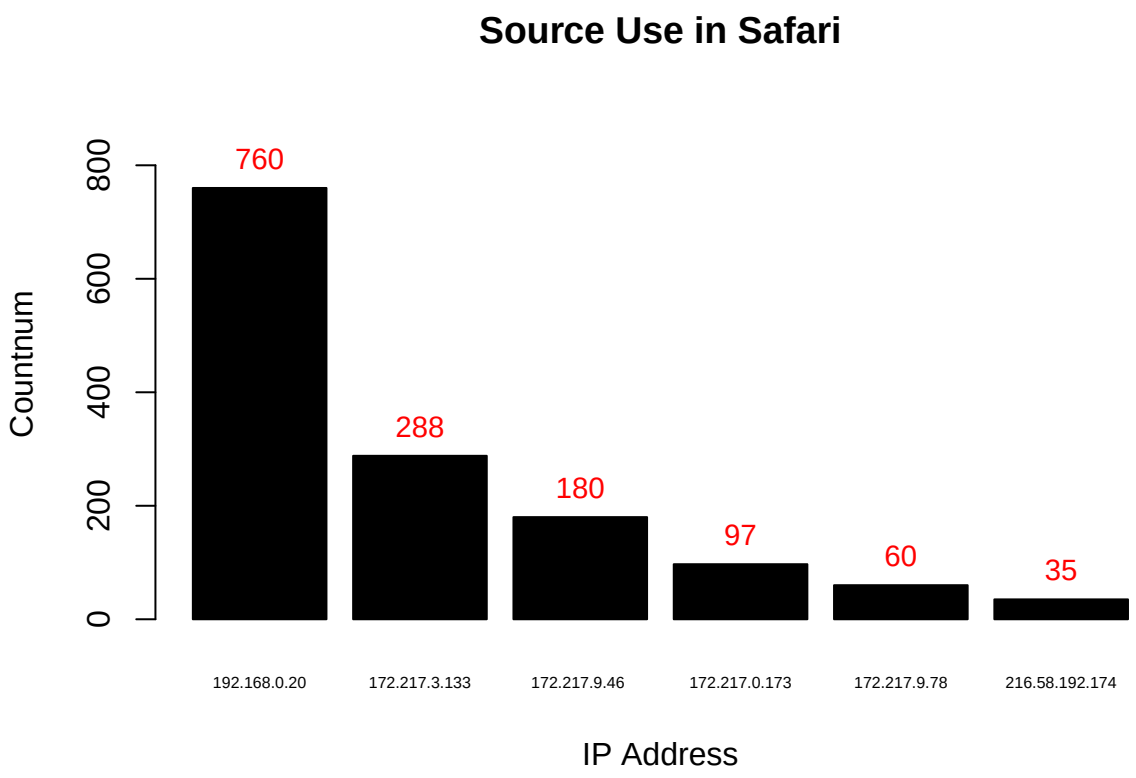
Top Hosts in Safari

```
library(readr)
hivesafari <- read_csv("C:/Users/Jsallese7/Desktop/hivesafari.csv")
ssource = hivesafari$Source
ssource1 = data.frame(ssource)
summary(ssource1)
```

```
##            ssource
##  192.168.0.20  :760
##  172.217.3.133 :288
##  172.217.9.46  :180
##  172.217.0.173 : 97
##  172.217.9.78  : 60
##  216.58.192.174: 35
##  (Other)       :118
```

```
sourcelist = c("192.168.0.20",
 "172.217.3.133",
 "172.217.9.46 ",
 "172.217.0.173",
 "172.217.9.78",
 "216.58.192.174")
sourcecount = c(760,288,180,97,60,35)
ss = barplot(sourcecount,
             col = "Black",
             main = "Source Use in Safari",
             xlab = "IP Address",
             ylab = "Countnum",
             ylim = range(0:900),
             names.arg = sourcelist,
```

```
            cex.names = .5)
text(x=ss,
     y=sourcecount,
     label = sourcecount,
     pos = 3,
     cex = 0.9,
     col = "red")
```



Above shows a bar plot of the top hosts found when the safari browser accessed gmail. Below is a link to an image of hive pulling and sorting the data found above.

Hive Code Source IP

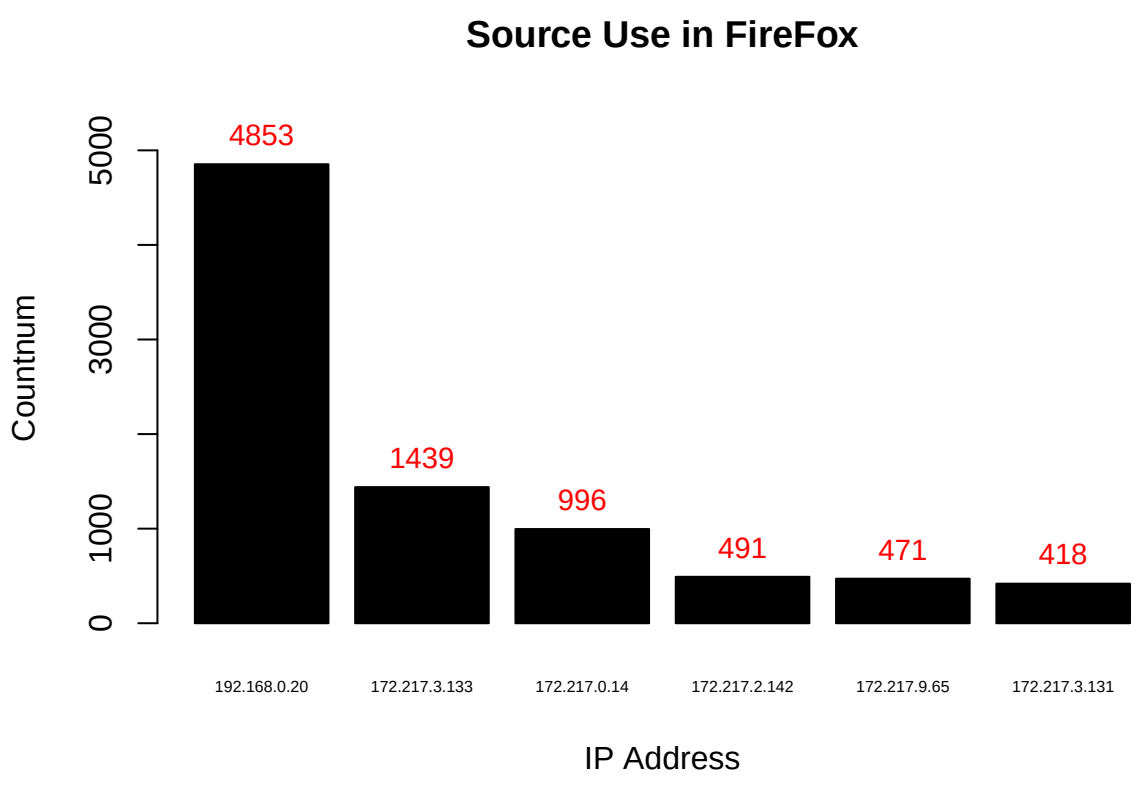
Top Hosts in Firefox

```
library(readr)
hivefirefox <- read_csv("C:/Users/Jsallese7/Desktop/hivefirefox.csv")
fsource = hivefirefox$Source
fsource1 = data.frame(fsource)
summary(fsource1)
```

```
##           fsource
##  192.168.0.20 :4853
##  172.217.3.133:1439
##  172.217.0.14 : 996
##  172.217.2.142: 491
```

```
##  172.217.9.65 : 471
##  172.217.3.131: 418
##  (Other)      :1678
```

```
fsourcelist = c("192.168.0.20",
 "172.217.3.133",
 "172.217.0.14",
 "172.217.2.142",
 "172.217.9.65" ,
 "172.217.3.131")
fsourcecount = c(4853,1439,996,491,471,418)
fs = barplot(fsourcecount,
             col = "Black",
             main = "Source Use in FireFox",
             xlab = "IP Address",
             ylab = "Countnum",
             ylim = range(0:5400),
             names.arg = fsourcelist,
             cex.names = .5)
text(x=fs,
     y=fsourcecount,
     label = fsourcecount,
     pos = 3,
     cex = 0.9,
     col = "red")
```



The above plot shows a plot of the top host used within the firefox browser while it was accessing gmail. By

comparing the safari and firefox plots one can infer that since there where more use of host on firefox safari sends and receives packets more efficiently.

Below is a link to an image of hive pulling and sorting the data found above.

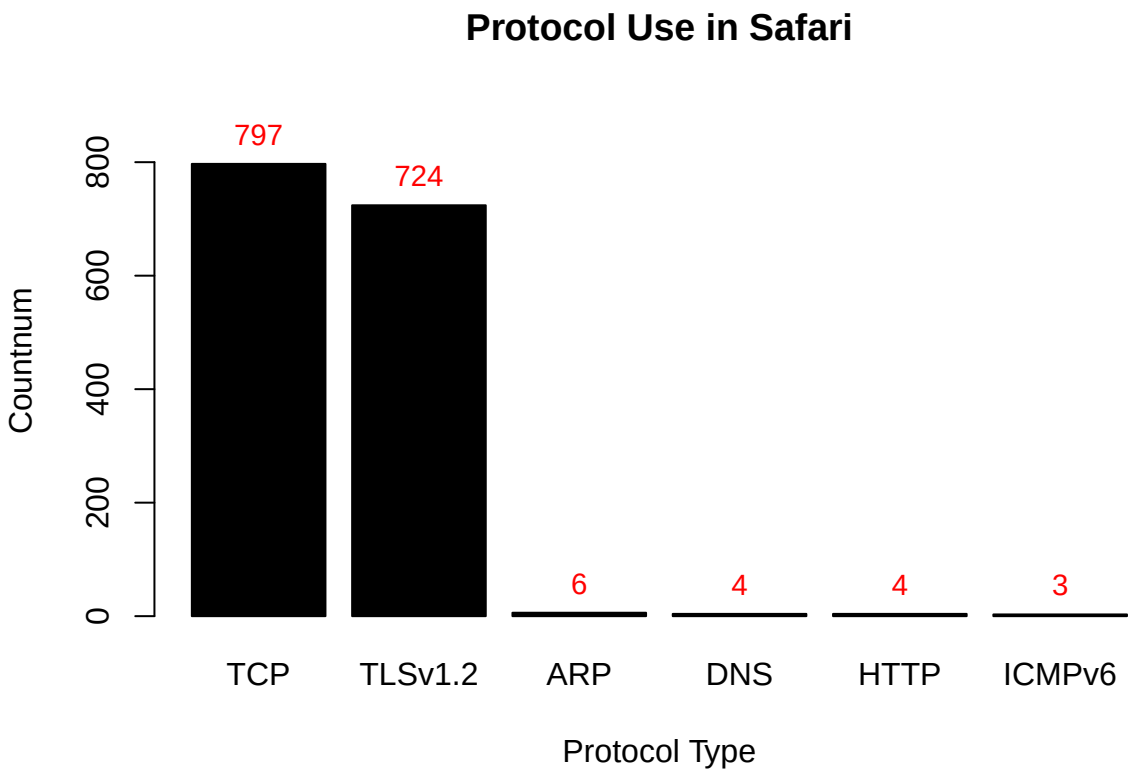
Hive Code Source IP

Protocol Use in Safari

```
library(readr)
hivesafari <- read_csv("C:/Users/Jsallese7/Desktop/hivesafari.csv")
proto12 = hivesafari$Protocol
prot = data.frame(proto12)
prot1 = summary(prot)
colnames(prot) <- c("Protocol")
listprotocol = c("TCP","TLSv1.2","ARP","DNS","HTTP","ICMPv6")
countnum = c(797,724,6,4,4,3)
prot1
```

```
##     proto12
##  ARP    :  6
##  DNS    :  4
##  HTTP   :  4
##  ICMPv6 :  3
##  TCP    :797
##  TLSv1.2:724
```

```
q = barplot(countnum,
            col = "Black",
            main = "Protocol Use in Safari",
            xlab = "Protocol Type",
            ylab = "Countnum",
            ylim = range(0:900),
            names.arg = listprotocol)
text(x=q,
     y=countnum,
     label = countnum,
     pos = 3,
     cex = 0.9,
     col = "red")
```



Above shows a plot of the protocol use in safari while gmail was accessed. As shown above TCP is used the most and TLS is used the second most and ICMPv6 is used the least.

Below is a link to an image of hive pulling and sorting the data found above.

Hive Code Protocol Use

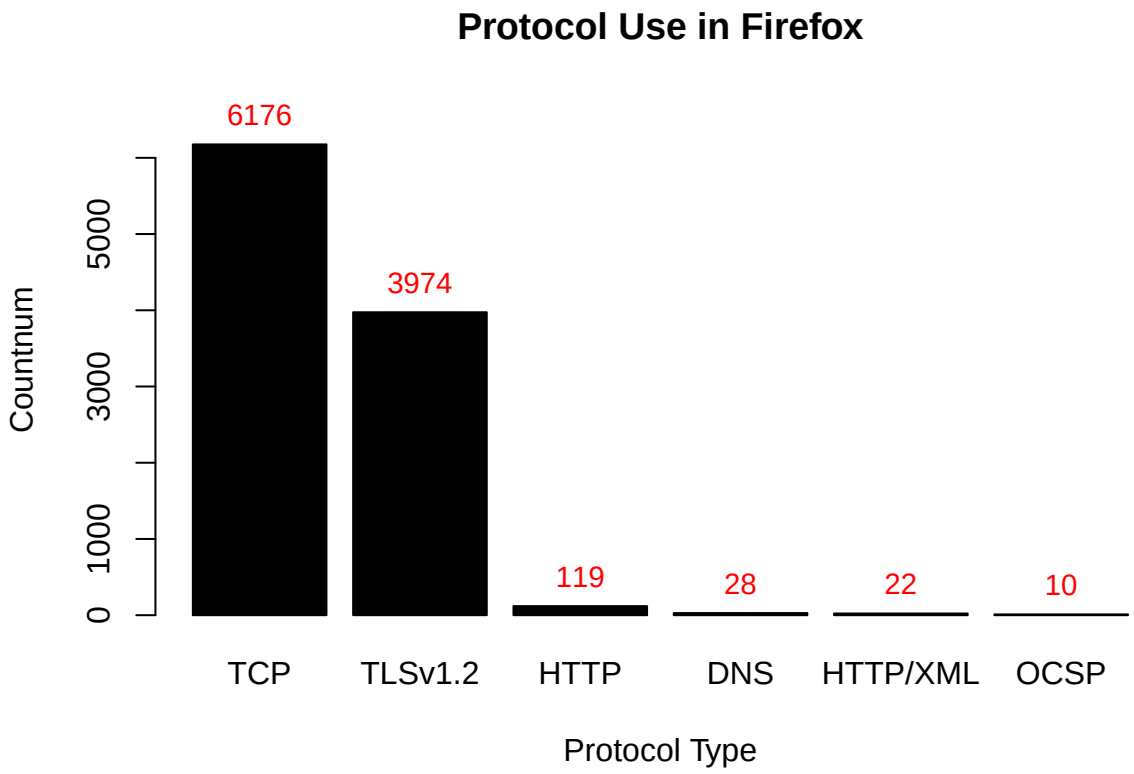
Protocol Use in Firefox

```
library(readr)
hivefirefox<- read_csv("C:/Users/Jsallese7/Desktop/hivefirefox.csv")

prot1 = hivefirefox$Protocol
prot2 = data.frame(prot1)
summary(prot2)
```

```
##       prot1
##  TCP     :6176
##  TLSv1.2 :3974
##  HTTP    : 119
##  DNS     :  28
##  HTTP/XML:  22
##  OCSP    :  10
##  (Other) :  17
```

```
colnames(prot2) <- c("Protocol")
protocollist = c("TCP", "TLSv1.2","HTTP","DNS","HTTP/XML","OCSP")
count = c(6176,3974,119,28,22,10)
q = barplot(count,
            col = "Black",
            main = "Protocol Use in Firefox",
            xlab = "Protocol Type",
            ylab = "Countnum",
            ylim = range(0:6700),
            names.arg = protocollist)
text(x=q,
     y=count,
     label = count,
     pos = 3,
     cex = 0.9,
     col = "red")
```



Shown above is a plot on the protocol use of firefox. Within the plot shown above like in the safar bar graph the TCP and the TLSv1.2 useage are the first and second in both.Comparing the two plots it is interesting to see that firefox has more protocol use than safari.

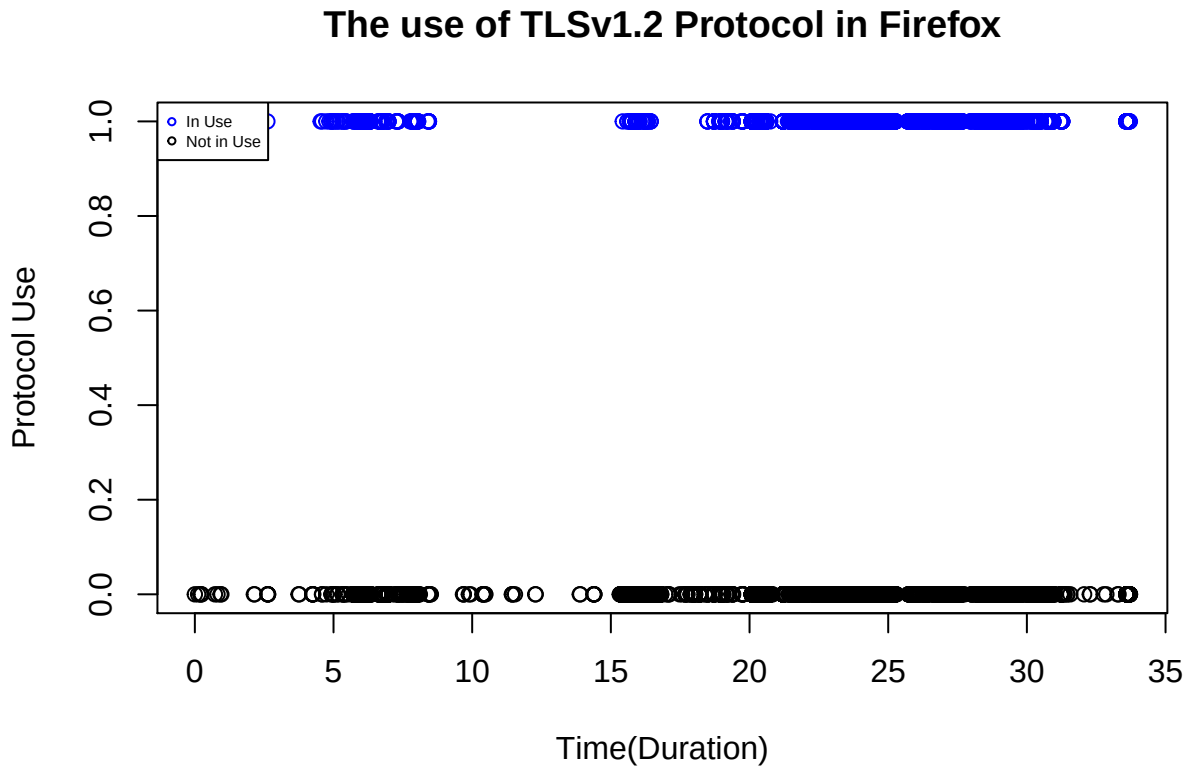
Hive Code Protocol Use

Transport Layer Security over Time in Firfox

```
library(readr)
hivefirefox <- read_csv("C:/Users/Jsallese7/Desktop/hivefirefox.csv")
ftime = hivefirefox$Time

library(stringr)
tlsuse = str_count(hivefirefox$Protocol, "TLSv1.2")


plot(ftime, tlsuse, main = "The use of TLSv1.2 Protocol in Firefox", xlab =
       "Time(Duration)",
ylab = "Protocol Use", pch = 1, col = ifelse(tlsuse == 1, "Blue", "Black"))
legend("topleft", c("In Use", "Not in Use"), pch = c(1,1), col=c("Blue", "Black"),
bg = "White", cex = 0.5)
```

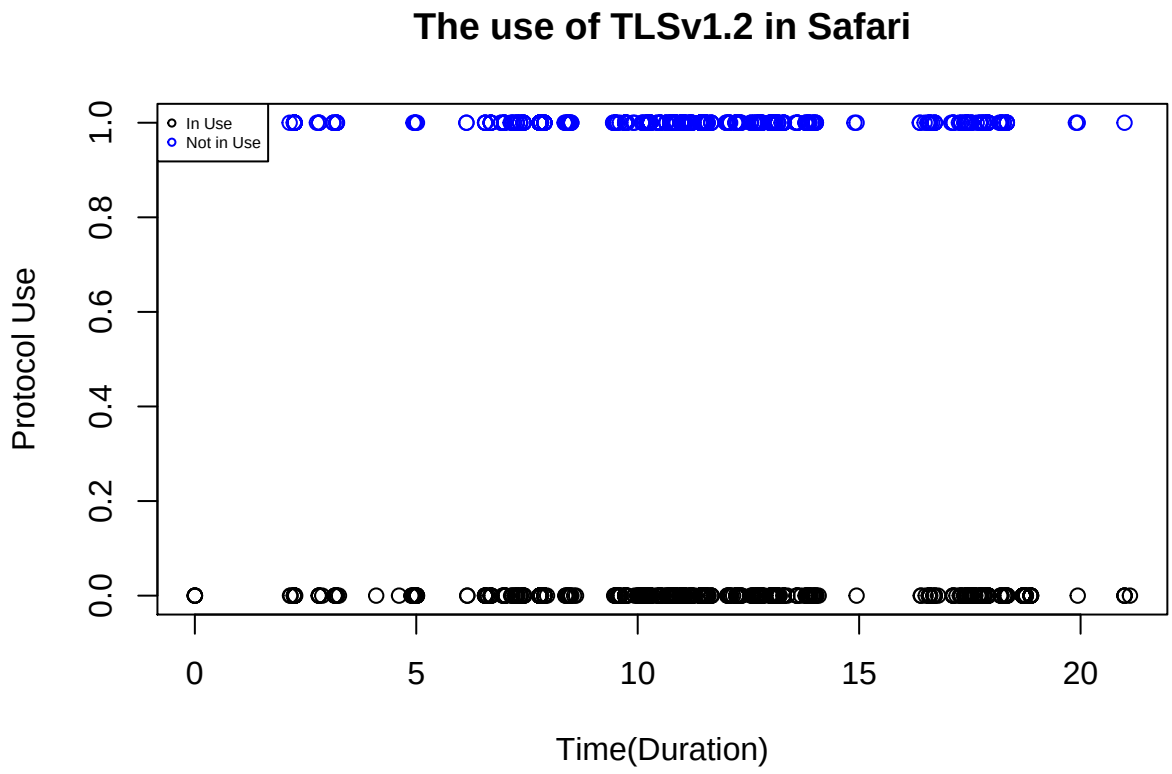


Shown above is a plot od the useage of the encryption protocol TLSv1.2 vs no usage over the duration that the data was captured in the firefox browser while it is accessing gmail.

```
library(readr)
hivesafari <- read_csv("C:/Users/Jsallese7/Desktop/hivesafari.csv")
stime = hivesafari$Time

library(stringr)
tlsuse1 = str_count(hivesafari$Protocol, "TLSv1.2")
```

```r
plot(stime,
     tlsuse1,
     main = "The use of TLSv1.2 in Safari",
     xlab = "Time(Duration)",
     ylab = "Protocol Use",
     pch = 1,
     col = ifelse(tlsuse1 == 1, "Blue", "Black"))
legend("topleft",
       c("In Use", "Not in Use"),
       pch = c(1,1),
       col=c("Black", "Blue"),
       bg = "White",
       cex = 0.5)
```



Shown above is a plot of the useage of the encryption protocol TLSv1.2 vs no usage in the safari browser while it is accessing gmail. Comparing the two plots above one can infer that safari is more secure because it the plot shows that it uses encryption more that firefox for the duration that packets where captured.

Types of packets in Safari

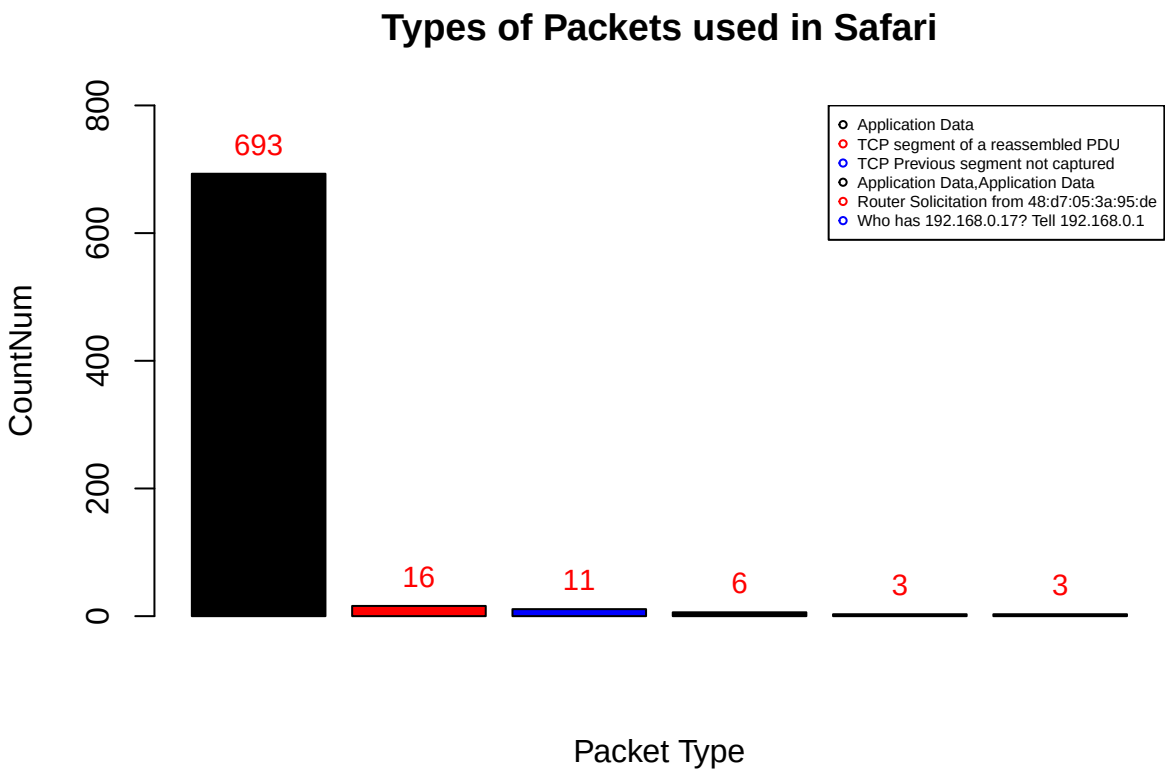
```r
library(readr)
hivesafari <- read_csv("C:/Users/Jsallese7/Desktop/hivesafari.csv")
sinfo = head(summary(factor(hivesafari$Info)))
sinfo11 = data.frame(sinfo)
```

```
colnames(sinfo11) <- c("Count")
sinfo11
```

```
##                                                        Count
## Application Data                                         693
## [TCP segment of a reassembled PDU]                        16
## [TCP Previous segment not captured] Application Data      11
## Application Data, Application Data                         6
## Router Solicitation from 48:d7:05:3a:95:de                 3
## Who has 192.168.0.17? Tell 192.168.0.1                     3
```

```
q5 = barplot(sinfo11$Count,
             col=c("Black", "Red","Blue"),
             main = "Types of Packets used in Safari",
             xlab = "Packet Type",
             ylab = "CountNum",
             ylim = range(0:800))
text(x=q5,
     y=sinfo11$Count,
     label = sinfo11$Count,
     pos = 3,
     cex = 0.9,
     col = "red")
legend("topright",
       c("Application Data",
         "TCP segment of a reassembled PDU",
         "TCP Previous segment not captured",
         "Application Data,Application Data",
         "Router Solicitation from 48:d7:05:3a:95:de",
         "Who has 192.168.0.17? Tell 192.168.0.1"),
       pch = c(1,1),
       col=c("Black", "Red","Blue"),

       bg = "White",
       cex = 0.5)
```



Above shows a bar plot of the packet types used in the safari browser while logging onto gmail.

Hive Packet Type

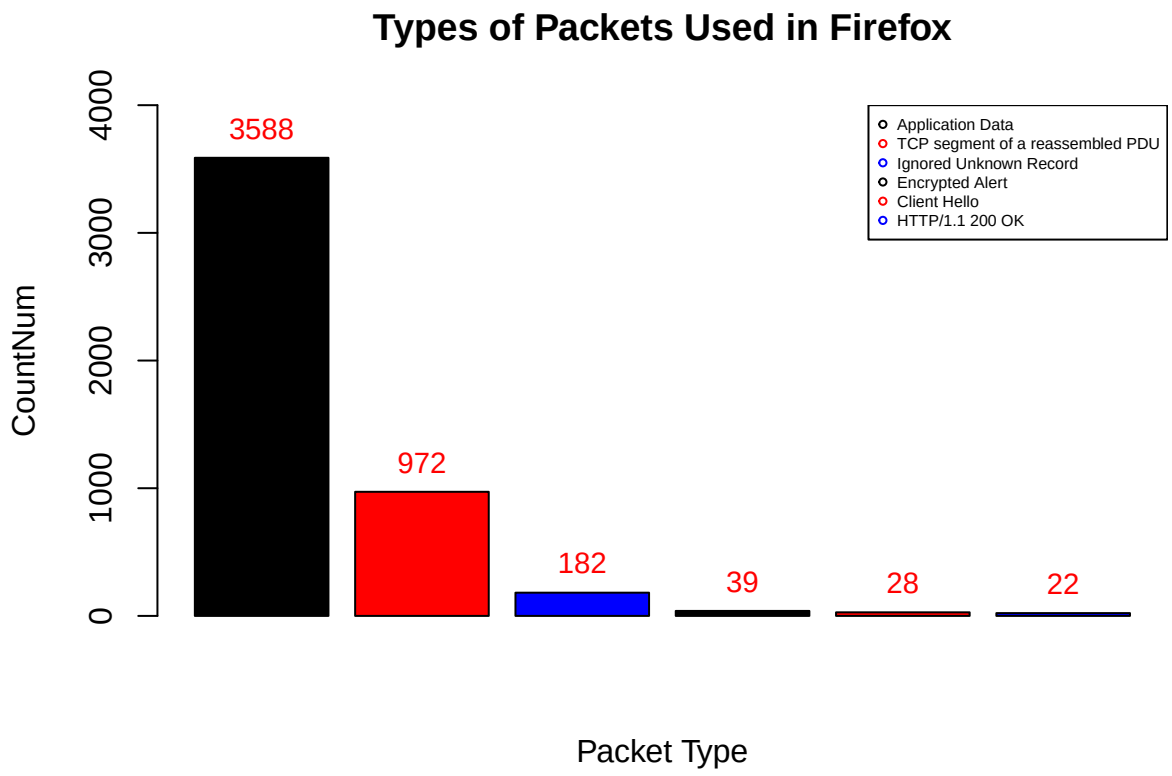
Types of packets in Firefox

```
library(readr)
hivefirefox <- read_csv("C:/Users/Jsallese7/Desktop/hivefirefox.csv")
finfo = head(summary(factor(hivefirefox$Info)))
finfo1 = data.frame(finfo)

colnames(finfo1) <- c("CountNum")

y = barplot(finfo1$CountNum,
            col=c("Black", "Red","Blue"),
            main = "Types of Packets Used in Firefox",
            xlab = "Packet Type",
            ylab = "CountNum",
            ylim = range(0:4000))
text(x=y,
     y=finfo1$CountNum,
     label = finfo1$CountNum,
     pos = 3,
     cex = 0.9,
     col = "red")
```

```
legend("topright",
c("Application Data",
"TCP segment of a reassembled PDU",
"Ignored Unknown Record",
"Encrypted Alert",
"Client Hello","HTTP/1.1 200 OK"),
       pch = c(1,1),
       col=c("Black", "Red","Blue"),
       bg = "White",
       cex = 0.5)
```



Above shows a bar plot of the packet type used in firefox. Comparing the two plots of packet type from safari and firfox I have come to the conclusion that according to the data shown in the plots the safari browser uses less packets overall so therefore as the data has shown the safari browser is overall more efficient that firefox.

Hive Packet Type

Conclusion: Throught the analysis I have learned more about Hive coding in cloudera and R coding in the R language.Within the future the usage of the cloudera VM to help sort dat will be incredibly usefull.