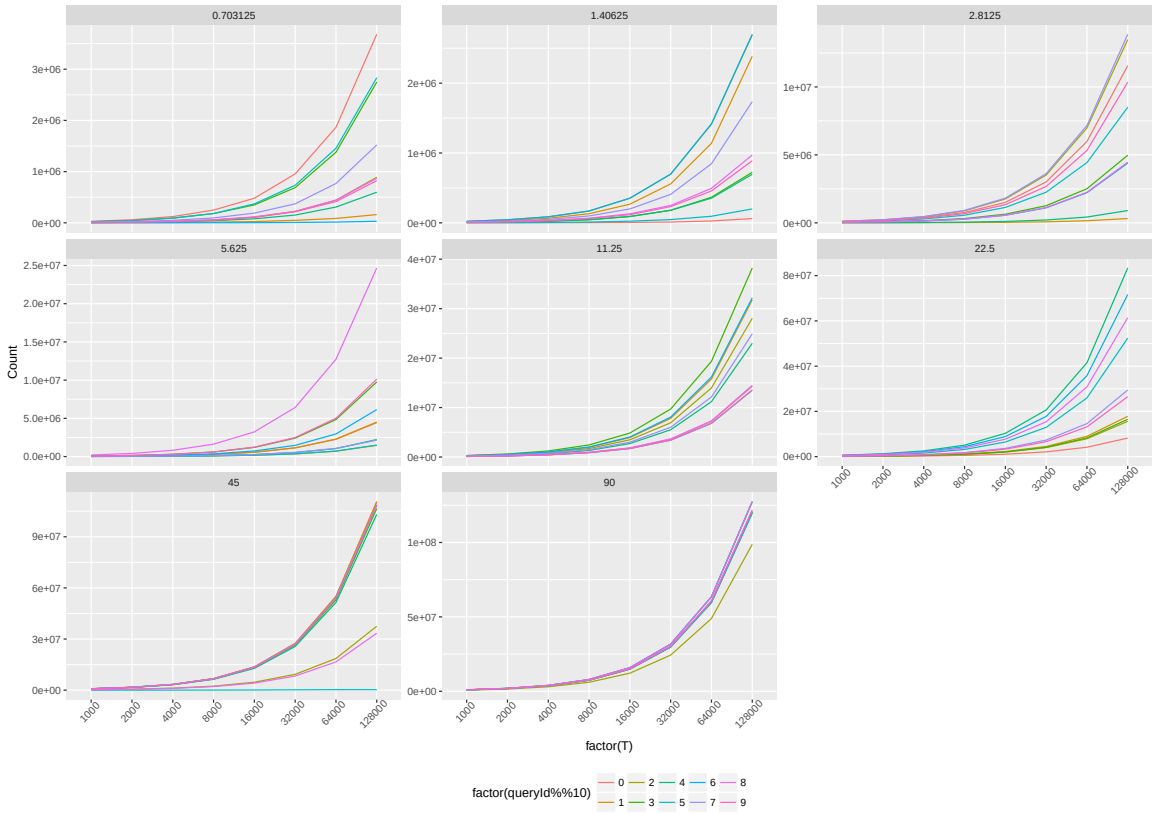
Results

October 17, 2017

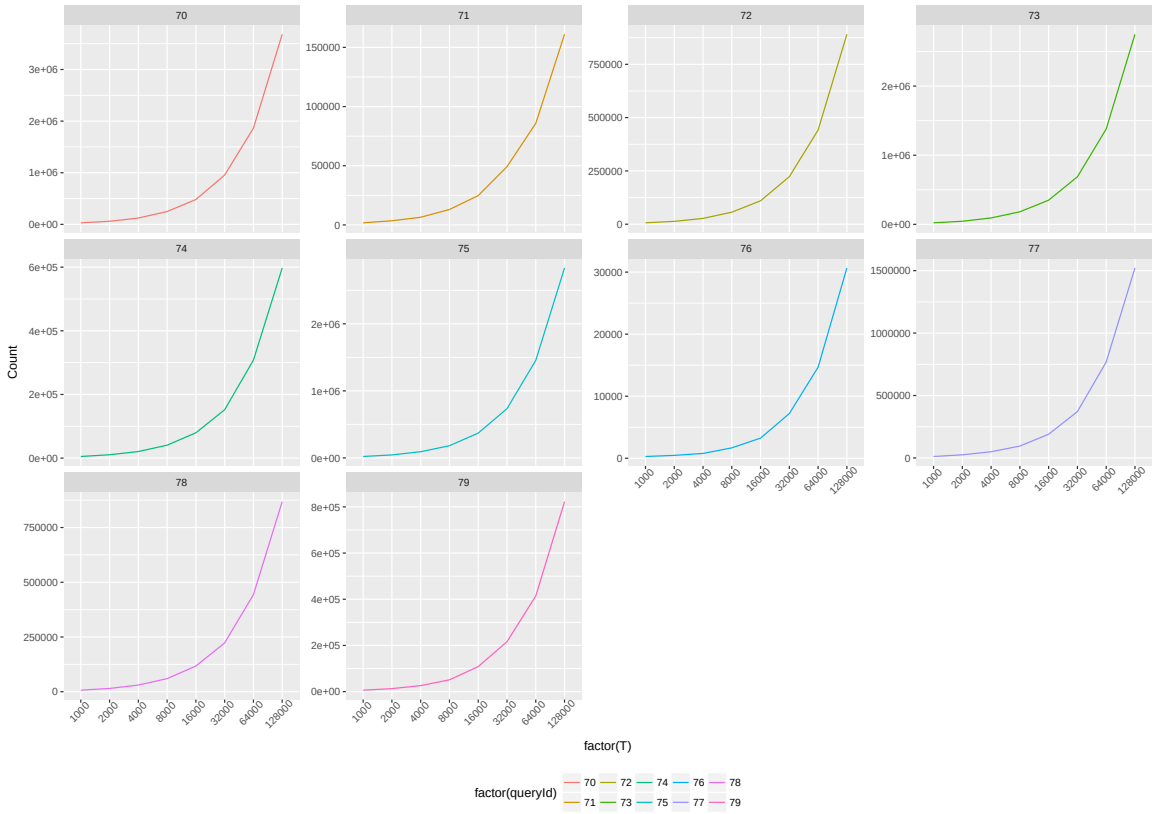
Contents

1 Plot: change of query count with size of T.

```
dfAvg %>%
    filter(EltSize == 16) %>%
    filter(algo == "BTree" & bench == "apply_at_region") %>%
    ggplot(aes(x = factor(T), y = Count, color = factor(queryId%%10))) +
    geom_line(aes(group=queryId))+
    facet_wrap(~queryWidth, scale = "free_y") +
    theme(legend.position = "bottom",
          axis.text.x = element_text(angle = 45, hjust = 0.75))
```

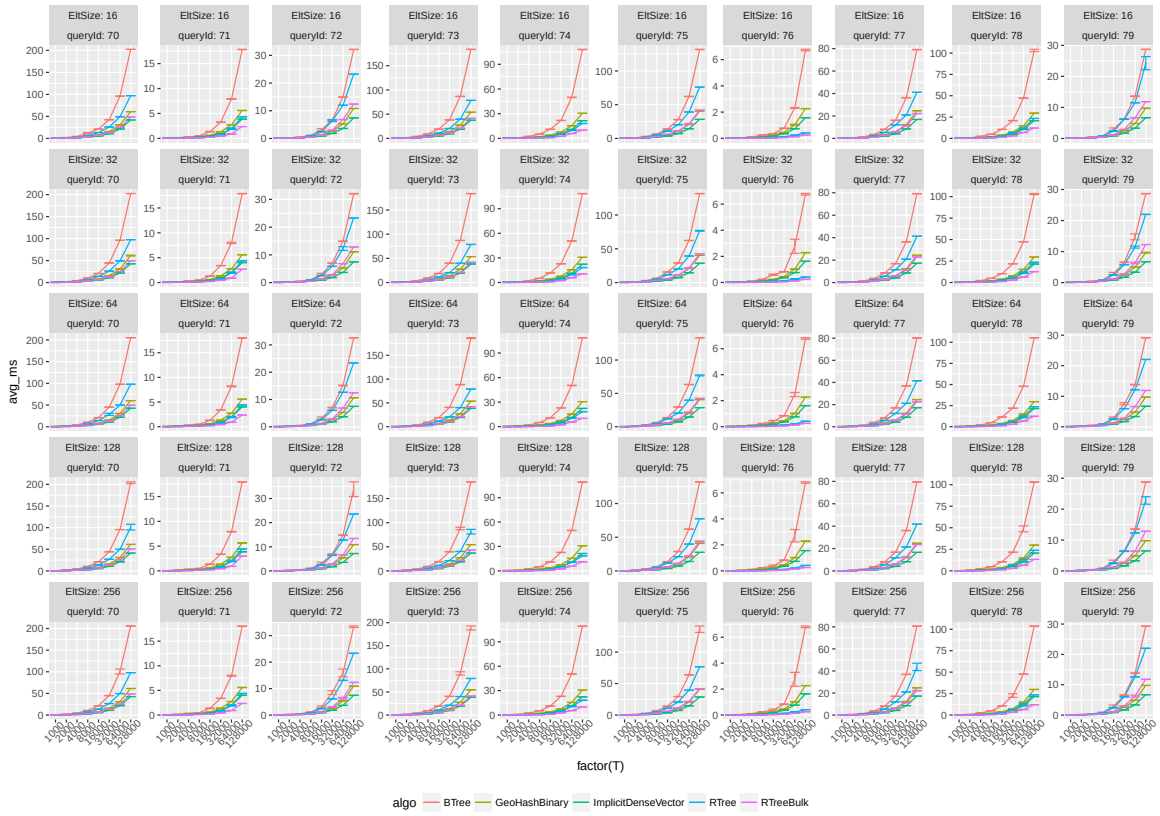



```
dfAvg %>%
    filter(EltSize == 16) %>%
    filter(algo == "BTree" & bench == "apply_at_region" & queryWidth < 1) %>%
    ggplot(aes(x = factor(T), y = Count, color = factor(queryId))) +
    geom_line(aes(group=queryId))+
    facet_wrap(~queryId, scale = "free_y") +
    theme(legend.position = "bottom",
          axis.text.x = element_text(angle = 45, hjust = 0.75))
```



2 Plot: Scan Query Time by T facet by queryId

```
dfAvg %>%
#    filter(EltSize == 0) %>%
    filter(bench == "scan_at_region" & queryWidth < 1) %>%
    ggplot(aes(x = factor(T), y = avg_ms, color = algo)) +
    geom_line(aes(group=algo))+
    geom_errorbar(aes(ymin = avg_ms - stdv , ymax = avg_ms + stdv)) +
    facet_wrap(EltSize~queryId, scale = "free_y",ncol=10,labeller=label_both) +
    theme(legend.position = "bottom",
          axis.text.x = element_text(angle = 45, hjust = 0.75))
```

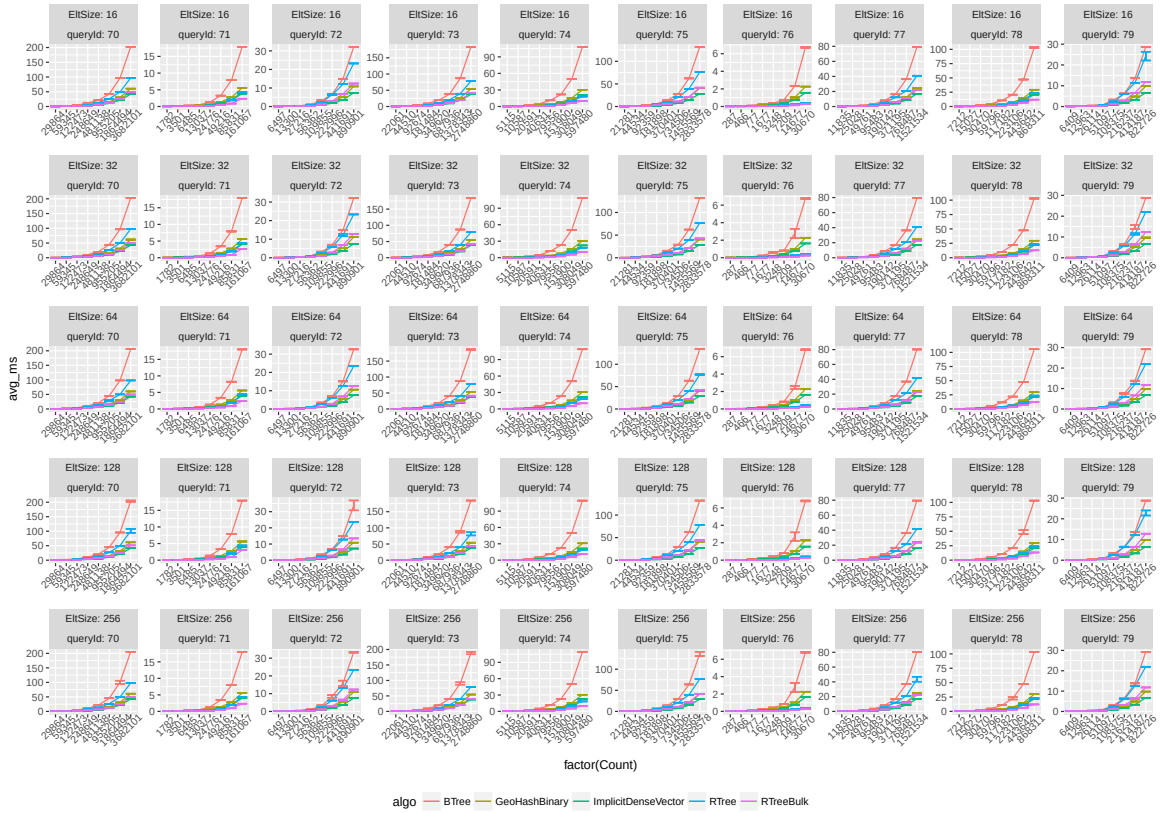


3 Plot: Scan query time by Query Count faceted by QueryId

```
dfAvg %>%
    ungroup() %>%
    filter(bench == "apply_at_region") %>%
    select(queryId,algo,Count,T) %>%
    left_join(
        filter(ungroup(dfAvg), bench == "scan_at_region") %>% select(-Count)
   ) -> dfCount
dfCount
```

```
Joining, by = c("queryId", "algo", "T")
# A tibble: 80,000 x 10
   queryId  algo  Count     T          bench Refine EltSize queryWidth   avg_ms        stdv
     <int> <chr>  <int> <int>          <chr>  <int>   <dbl>      <dbl>    <dbl>       <dbl>
 1       0 BTree 924827  1000 scan_at_region     29      16         90 17.43356 0.010635078
 2       0 BTree 924827  1000 scan_at_region     29      32         90 17.48158 0.022587843
 3       0 BTree 924827  1000 scan_at_region     29      64         90 17.50776 0.017892407
 4       0 BTree 924827  1000 scan_at_region     29     128         90 17.48583 0.008840569
 5       0 BTree 924827  1000 scan_at_region     29     256         90 17.61844 0.032610912
 6       0 BTree 924827  1000 scan_at_region     29      16         90 17.43356 0.010635078
 7       0 BTree 924827  1000 scan_at_region     29      32         90 17.48158 0.022587843
 8       0 BTree 924827  1000 scan_at_region     29      64         90 17.50776 0.017892407
 9       0 BTree 924827  1000 scan_at_region     29     128         90 17.48583 0.008840569
10       0 BTree 924827  1000 scan_at_region     29     256         90 17.61844 0.032610912
# ... with 79,990 more rows
```

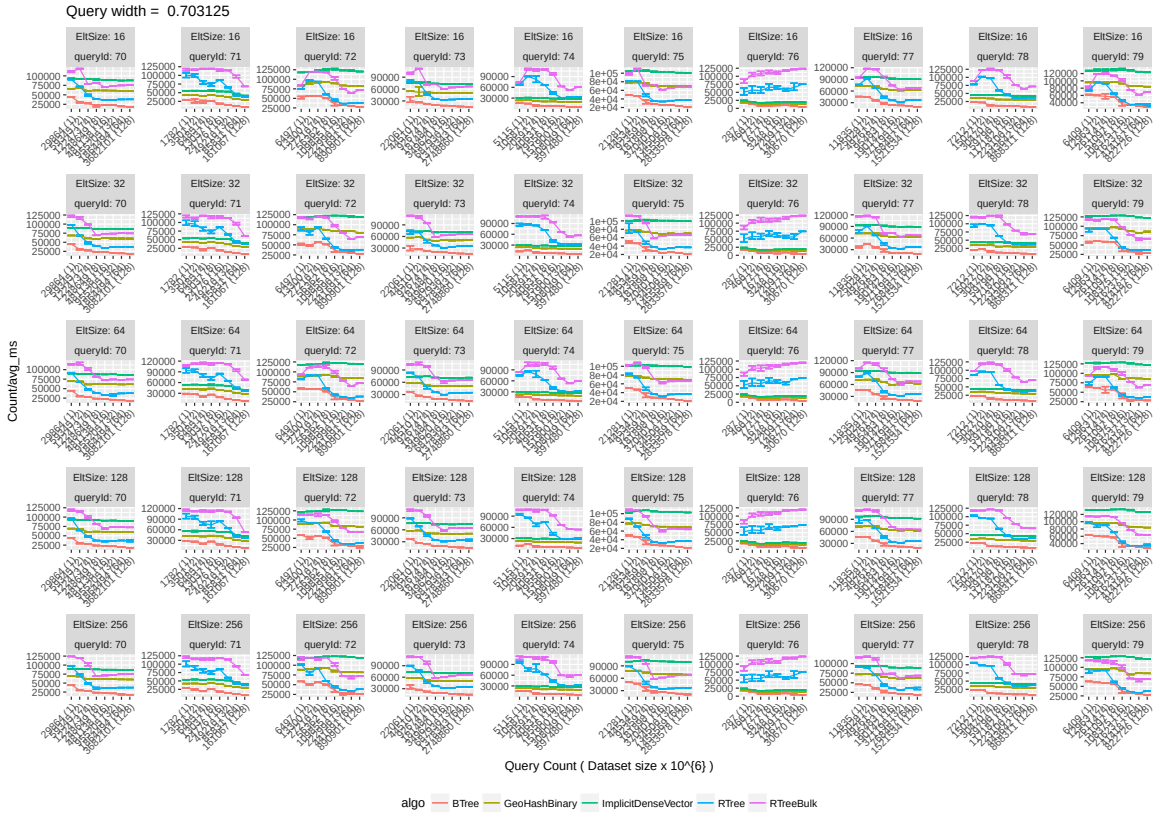
```
dfCount %>%
    filter(queryWidth < 1) %>%
    ggplot(aes(x = factor(Count), y = avg_ms, color = algo)) +
    geom_line(aes(group=algo))+
    geom_errorbar(aes(ymin = avg_ms - stdv , ymax = avg_ms + stdv)) +
    facet_wrap(EltSize~queryId, scale = "free",ncol=10, labeller = label_both) +
    theme(legend.position = "bottom",
          axis.text.x = element_text(angle = 45, hjust = 0.75))
```

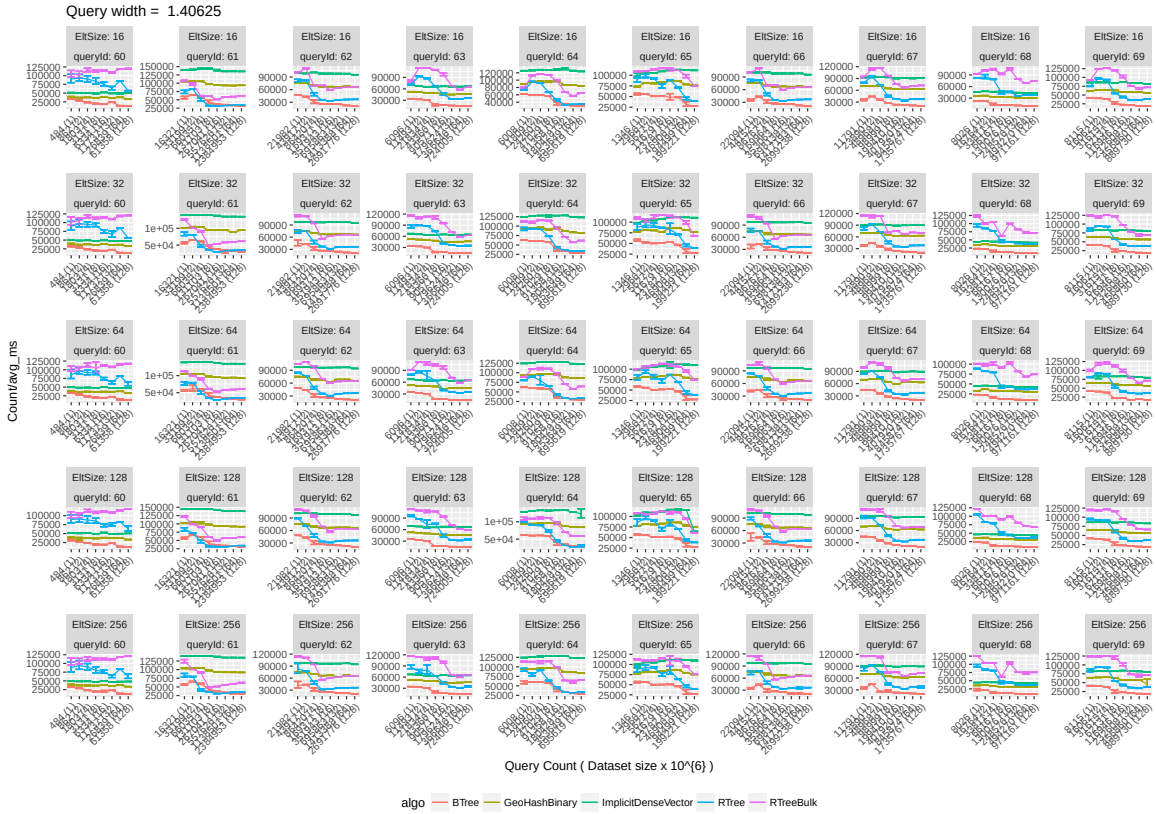


4 Plot: Scan query throughput by Query Count faceted by QueryId

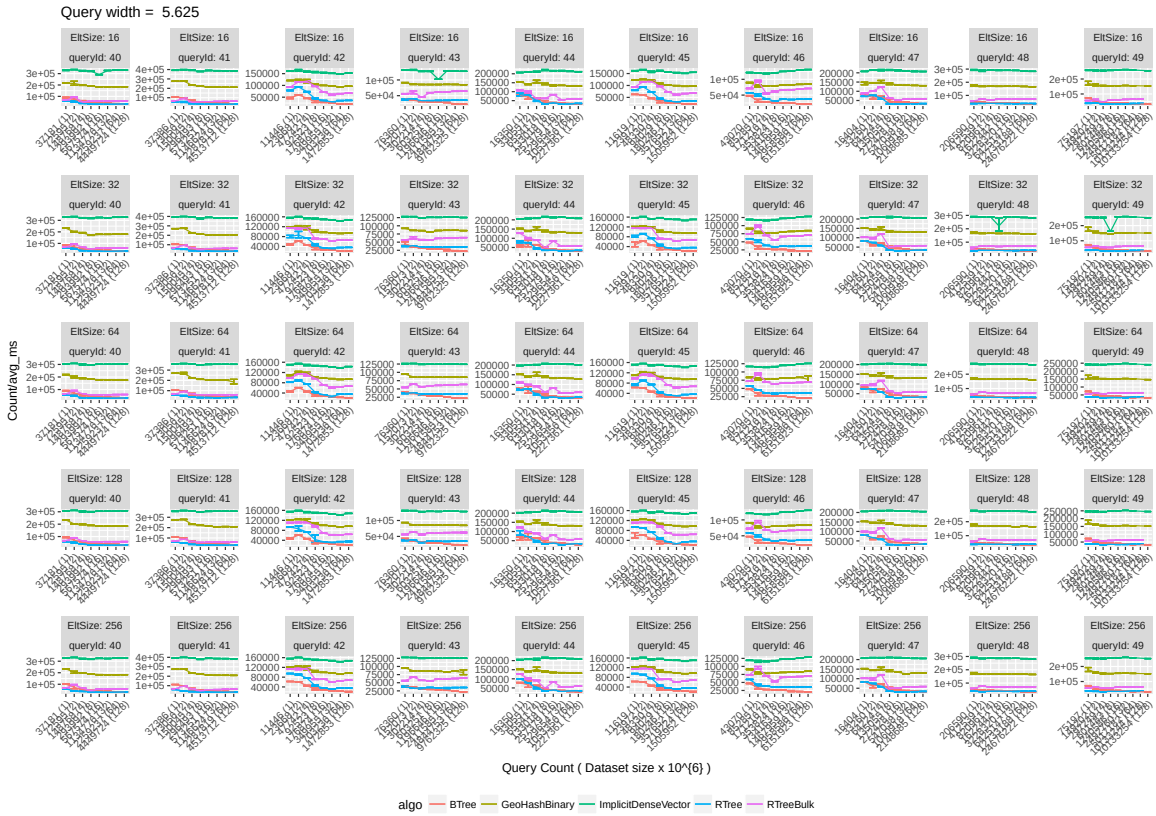
```
tgpPlot <- function(dfCount){
    dfCount %>%
        filter(queryWidth == w) %>%
        arrange(Count,T) %>%
        mutate(lbls = paste(Count," (",T/1000,")",sep="")) %>%
        ggplot(aes(x = factor(lbls,levels=unique(lbls)), group=algo, y = Count / avg_ms, color = algo))
                                        #ggplot(aes(x = factor(Count), group=algo, y = Count / avg_ms,
        geom_errorbar(aes(ymin = Count / (avg_ms - stdv) , ymax = Count / (avg_ms + stdv)) ) +
        geom_line() +
        facet_wrap(EltSize~queryId, scale = "free",ncol=10, labeller = label_both) +
        theme(legend.position = "bottom",
              axis.text.x = element_text(angle = 45, hjust = 1)) +
        labs(x = "Query Count ( Dataset size x 10^{6} )",
             title = paste("Query width = ", w))
```

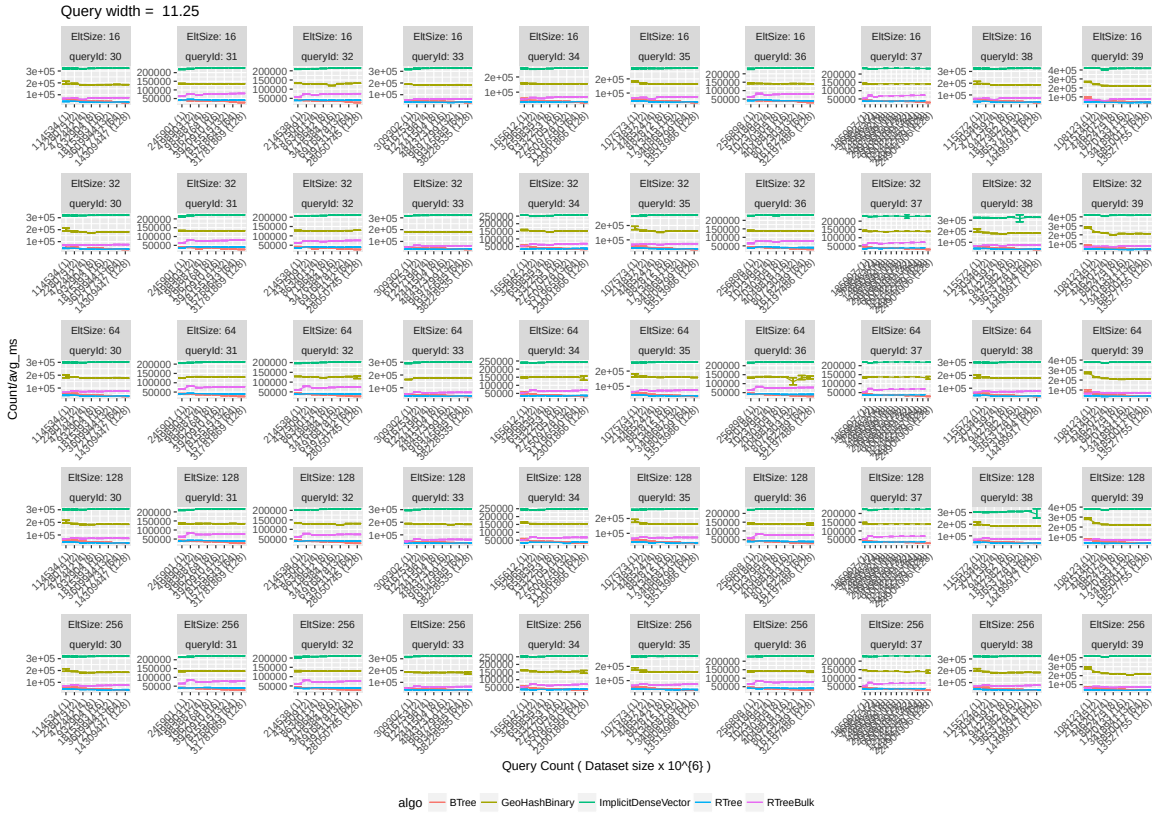
}

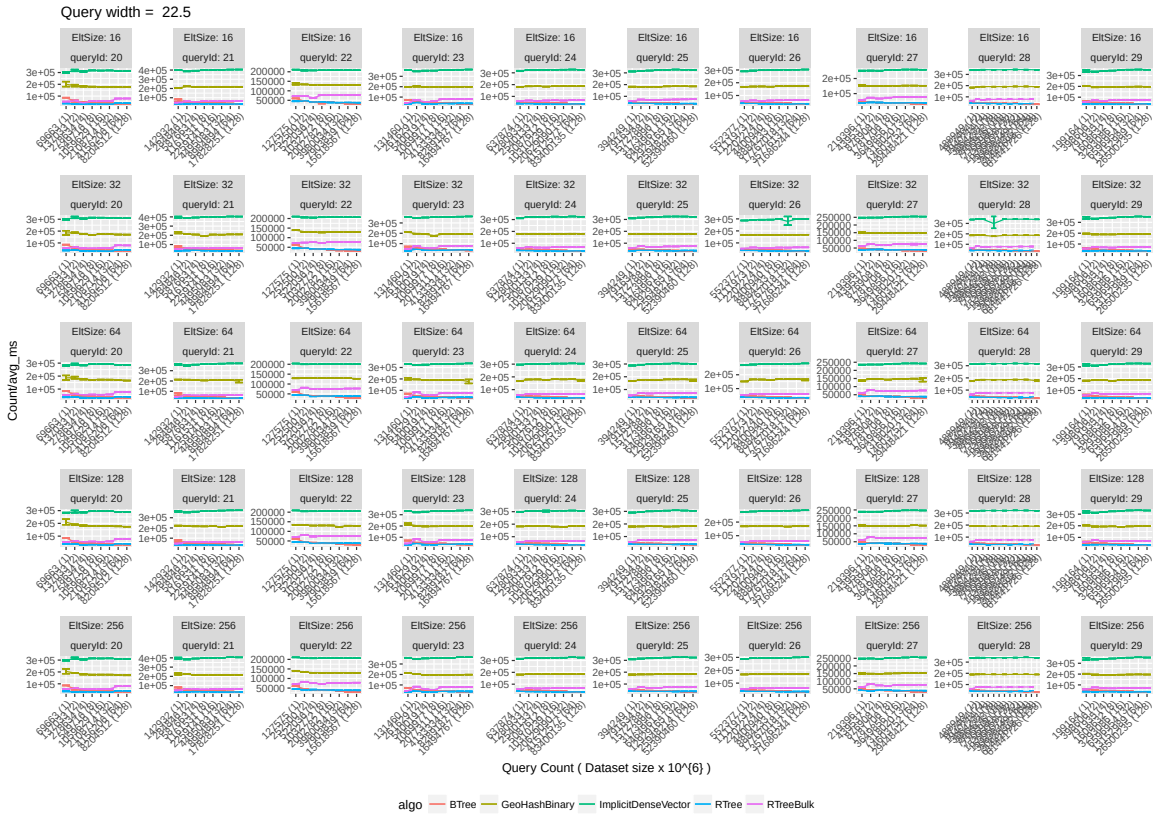

Query width = 0.703125
Count/avg_ms
Query Count (Dataset size x 10^{6})
algo BTree GeoHashBinary ImplicitDenseVector RTree RTreeBulk

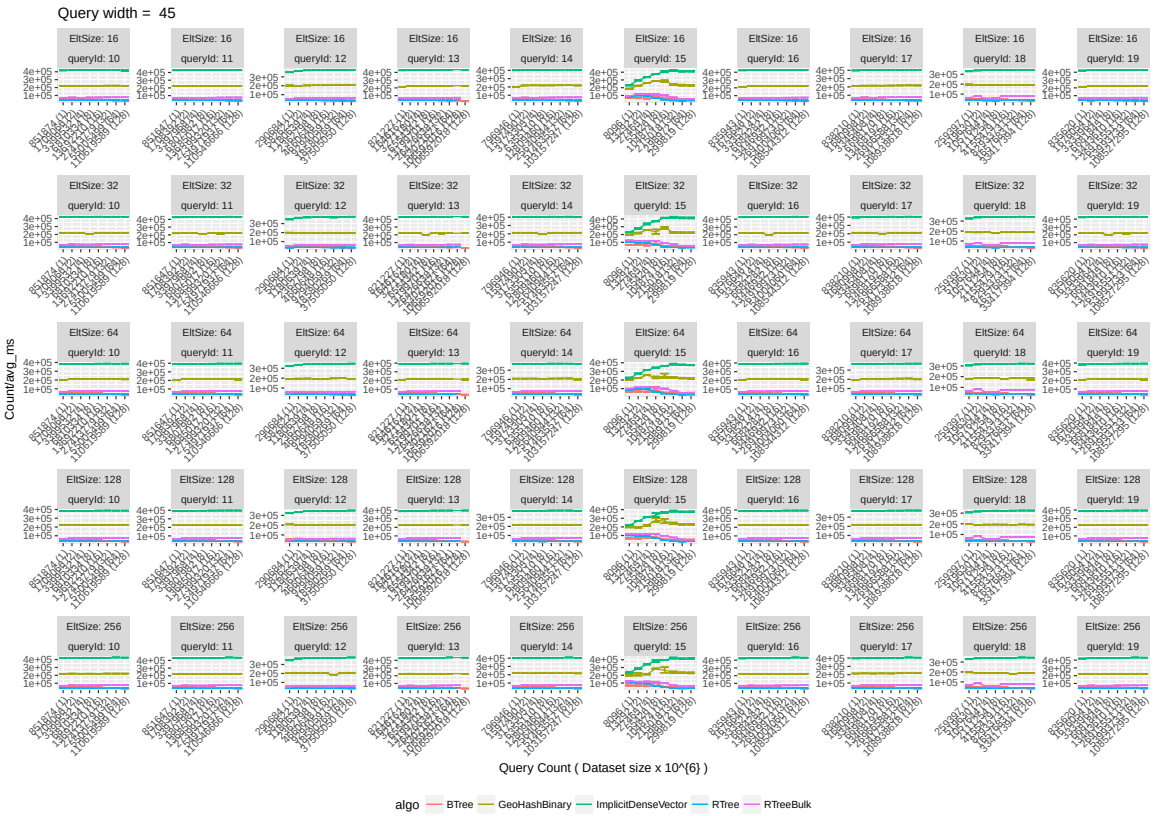

Query width = 1.40625
Count/avg_ms
Query Count (Dataset size x 10^{6})
algo BTree GeoHashBinary ImplicitDenseVector RTree RTreeBulk


Query width = 2.8125
Count/avg_ns
Query Count (Dataset size x 10^{6})
algo BTree GeoHashBinary ImplicitDenseVector RTree RTreeBulk


Query width = 5.625
Count/avg_ns
Query Count (Dataset size x 10^{6})
algo BTree GeoHashBinary ImplicitDenseVector RTree RTreeBulk


Query width = 11.25
Count/avg_ns
Query Count (Dataset size x 10^{6})
algo BTree GeoHashBinary ImplicitDenseVector RTree RTreeBulk


Query width = 22.5
Count/avg_ns
Query Count (Dataset size x 10^{6})
algo BTree GeoHashBinary ImplicitDenseVector RTree RTreeBulk


Query width = 45
EltSize: 16
queryId: 10
queryId: 11
queryId: 12
queryId: 13
queryId: 14
queryId: 15
queryId: 16
queryId: 17
queryId: 18
queryId: 19
EltSize: 32
EltSize: 64
EltSize: 128
EltSize: 256
Count/avg_ms
Query Count (Dataset size x 10^{6})
algo
BTree
GeoHashBinary
ImplicitDenseVector
RTree
RTreeBulk