Results Section: Public Sequencing Metrics

```
library(staphopia)
library(ggplot2)
library(reshape2)
USE_DEV = TRUE
```

Aggregating Data For Public Samples

First we'll get all publicly available *S. aureus* samples.

```
ps <- get_public_samples()
```

We will also get information pertaining to submissions and ranks by year.

```
submissions <- get_submission_by_year()
ranks <- get_rank_by_year()
```

We now have 42949 samples to work with. Next we will acquire metadata, sequencing stats and assembly stats associated with each sample.

```
metrics <- merge(
    ps,
    merge(
        get_assembly_stats(ps$sample_id),
        merge(
            get_metadata(ps$sample_id),
            get_sequence_quality(ps$sample_id, stage='cleanup'),
            by='sample_id'
        ),
        by='sample_id'
    ),
    by='sample_id'
)
```

We are now going to add two columns `rank_name` and `year`.

```
metrics$year <- sapply(
    metrics$first_public,
    function(x) {
        strsplit(x, "-")[[1]][1]
    }
)

metrics$rank_name <- ifelse(
    metrics$rank.x == 3,
    'Gold',
    ifelse(
        metrics$rank.x == 2,
        'Silver',
        'Bronze'
    )
)
```

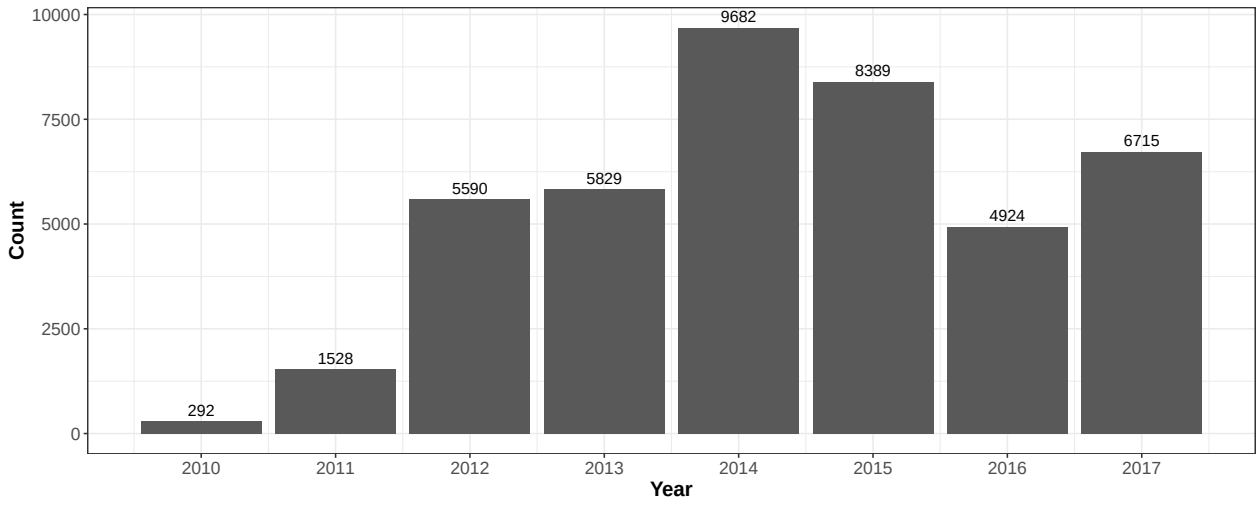
Visualizing Metrics

The following sections will be plots to visualize relationships in the data.

By Year Plots

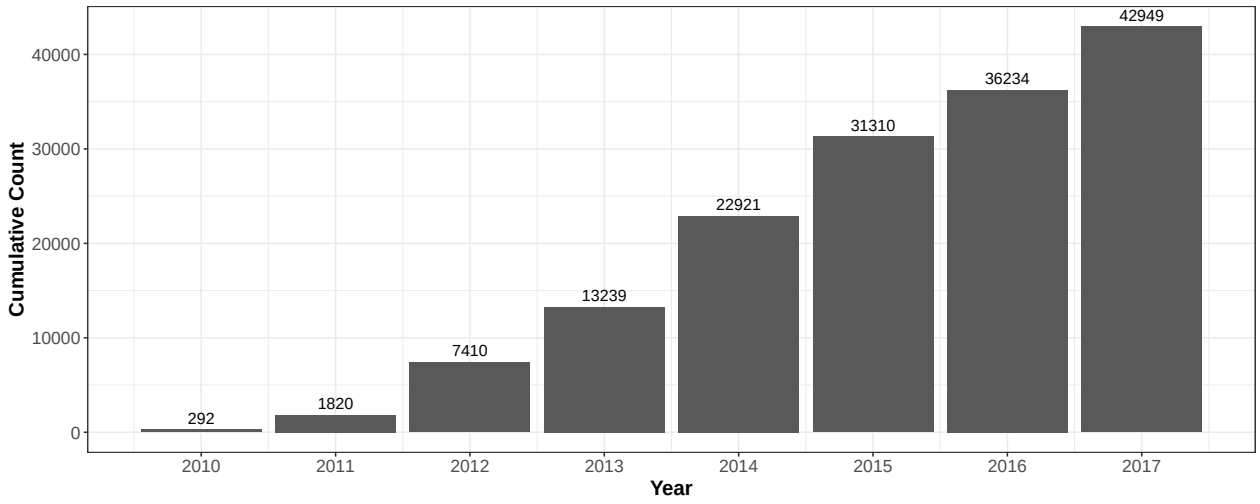
Submissions Per Year

```
p <- ggplot(data=submissions, aes(x=year, y=count)) +
    xlab("Year") +
    ylab("Count") +
    geom_bar(stat='identity') +
    geom_text(aes(label=count), vjust = -0.5) +
    scale_x_continuous(breaks = round(seq(min(submissions$year), max(submissions$year), by = 1),1)) +
    theme_bw() +
    theme(axis.text=element_text(size=12),
          axis.title=element_text(size=14,face="bold"))
p
```



Overall Submissions

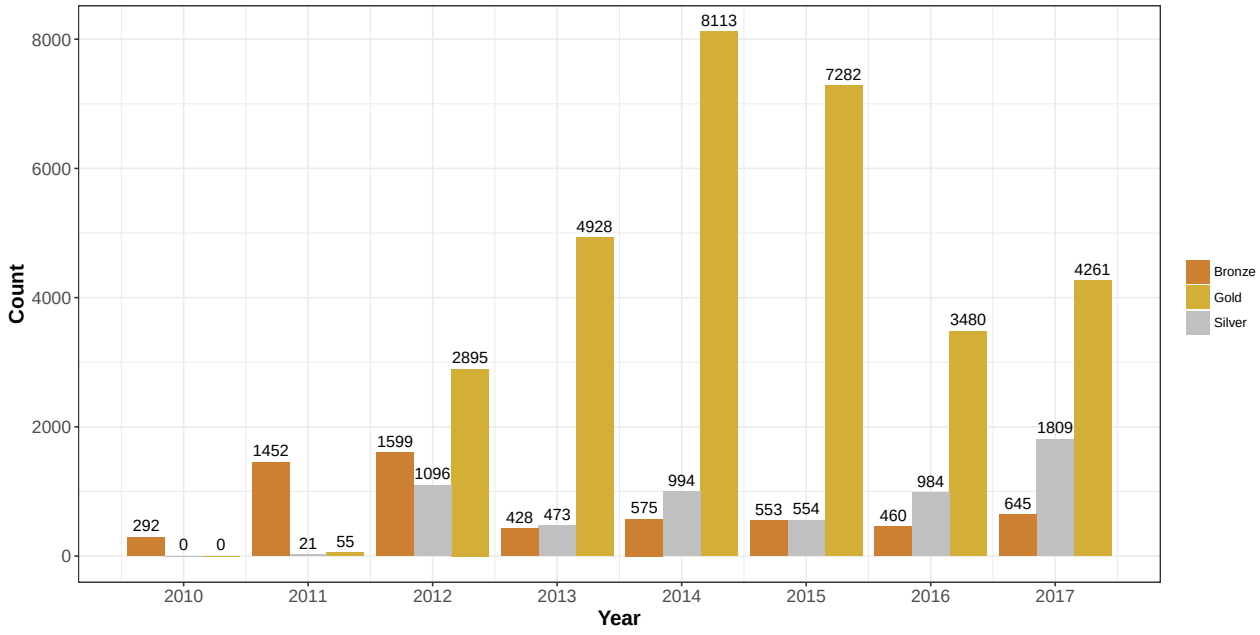
```
p <- ggplot(data=submissions, aes(x=year, y=overall)) +
    xlab("Year") +
    ylab("Cumulative Count") +
    geom_bar(stat='identity') +
    geom_text(aes(label=overall), vjust = -0.5) +
    scale_x_continuous(breaks = round(seq(min(submissions$year), max(submissions$year), by = 1),1)) +
    theme_bw() +
    theme(axis.text=element_text(size=12),
          axis.title=element_text(size=14,face="bold"))
p
```




```
# Output plot to PDF and PNG
staphopia::write_plot(p, paste0(getwd(), '/images/figure-x-submissions-per-year'))
```

Submission Ranks

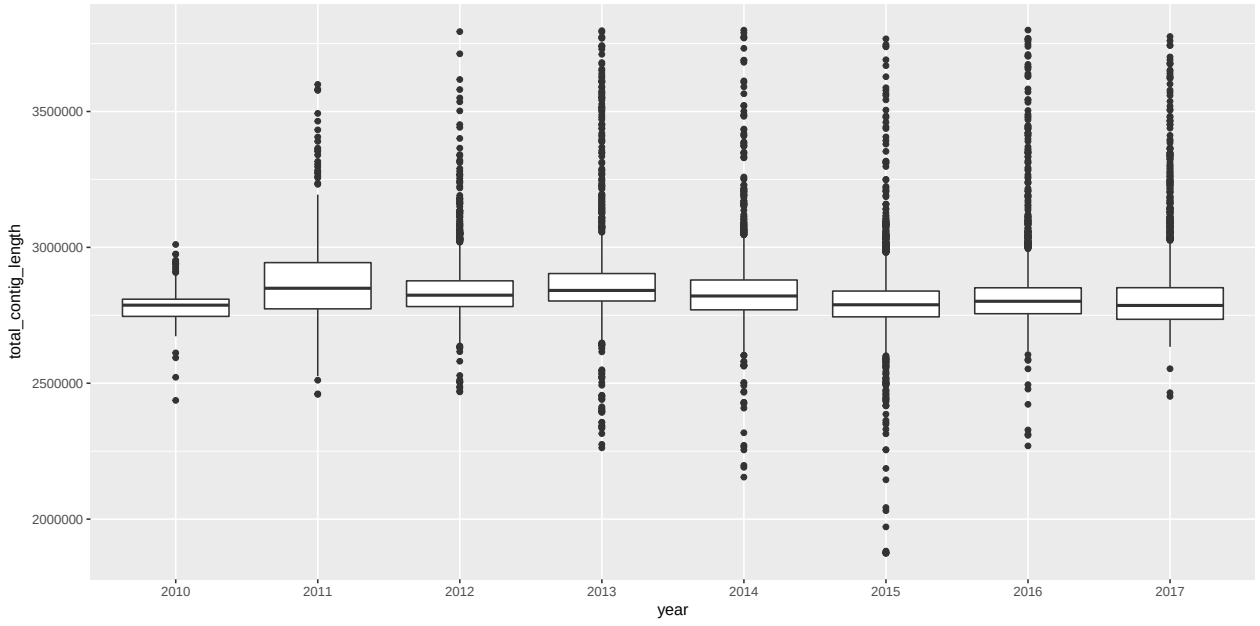
```
melted <- melt(ranks, id=c('year'),
               measure.vars = c('bronze', 'silver', 'gold'))
melted$title <- ifelse(melted$variable == 'gold', 'Gold',
                       ifelse(melted$variable == 'silver', 'Silver', 'Bronze'))
melted$rank <- ifelse(melted$variable == 'gold', 3,
                      ifelse(melted$variable == 'silver', 2, 1))
p <- ggplot(data=melted, aes(x=year, y=value, fill=title, group=rank, label=title)) +
    xlab("Year") +
    ylab("Count") +
    geom_bar(stat='identity', position='dodge') +
    geom_text(aes(label=value), vjust = -0.5, position = position_dodge(.9)) +
    scale_fill_manual(values=c("#CD7F32", "#D4AF37", "#C0C0C0")) +
    scale_x_continuous(breaks = round(seq(min(ranks$year), max(ranks$year), by = 1),1)) +
    theme_bw() +
    theme(axis.text=element_text(size=12),
          axis.title=element_text(size=14,face="bold"),
          legend.title = element_blank())


p
```




```
# Output plot to PDF and PNG
staphopia::write_plot(p, paste0(getwd(), '/images/figure-x-rank-per-year'))
```

Assembly Size

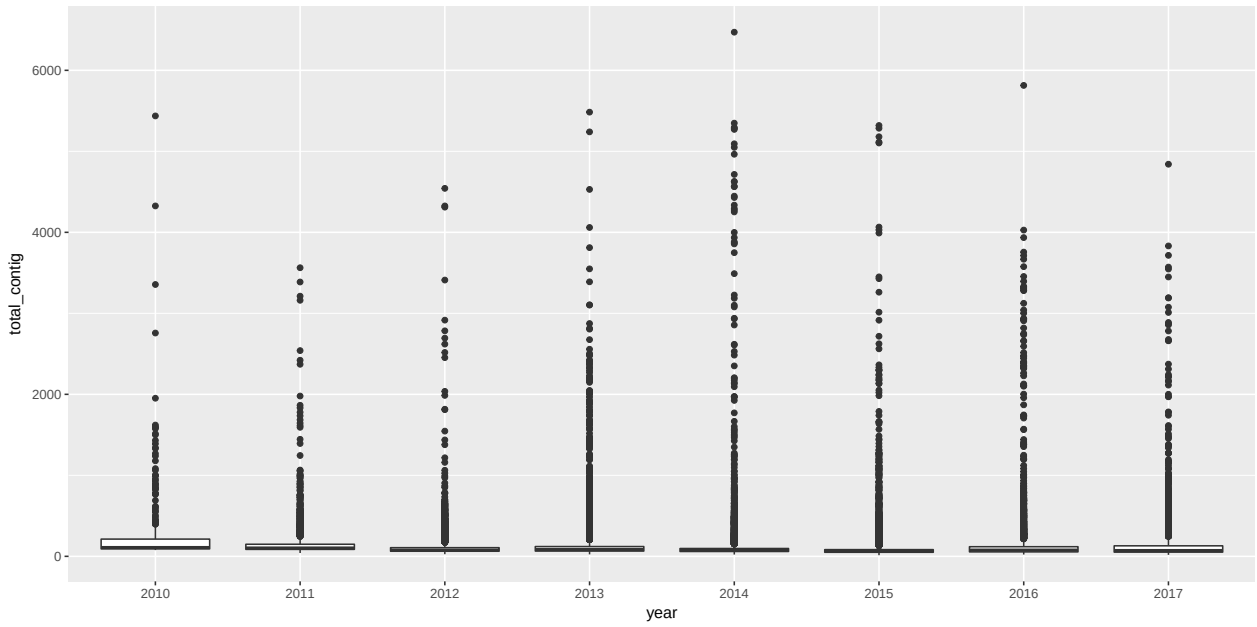
```
p <- ggplot(metrics, aes(x = year, y = total_contig_length)) +
    geom_boxplot()
p
```



Total Contigs (smaller is better)

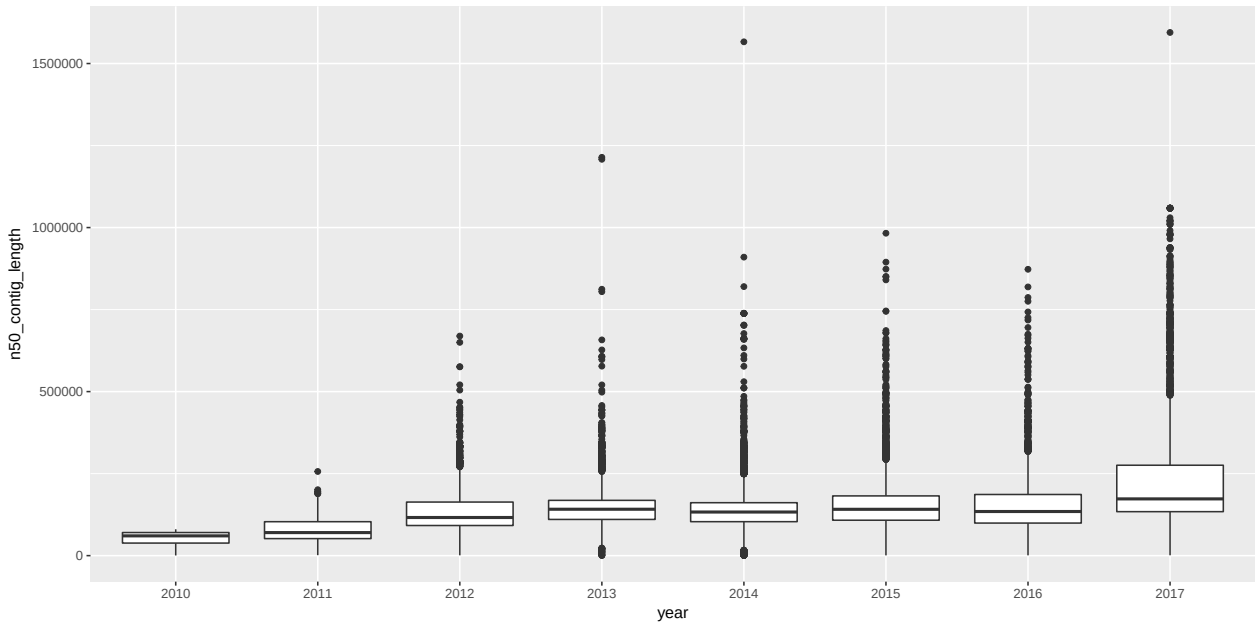
```
p <- ggplot(metrics, aes(x = year, y = total_contig)) +
    geom_boxplot()
```

```
p
```



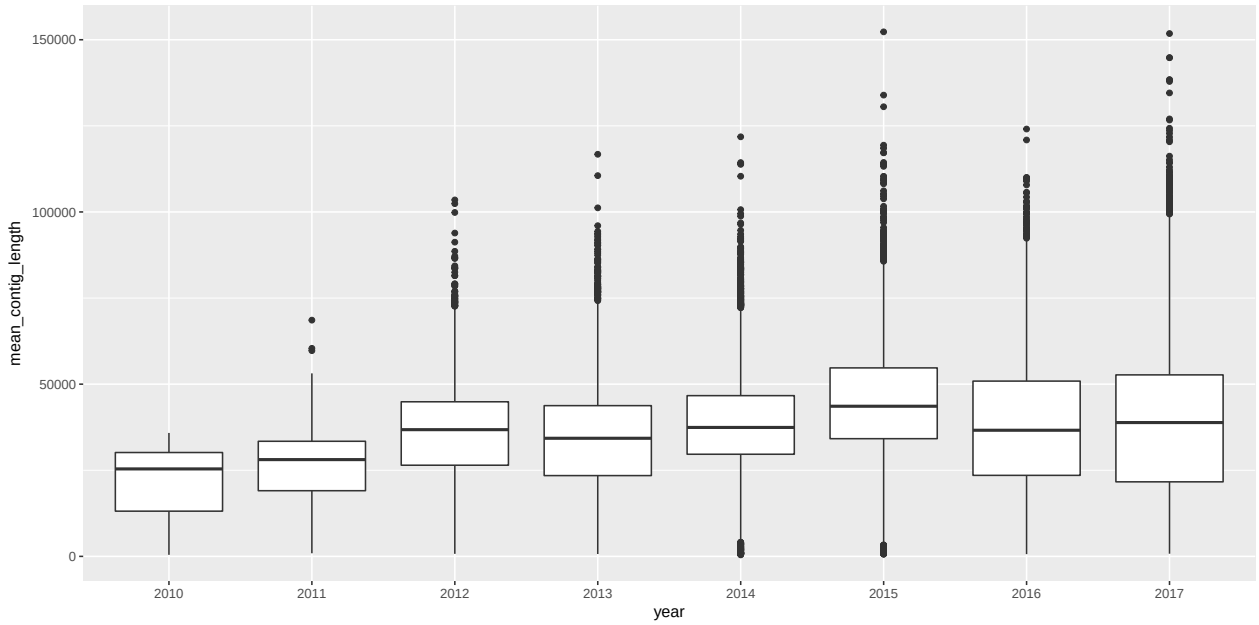
N50

```
p <- ggplot(metrics, aes(x = year, y = n50_contig_length)) +
    geom_boxplot()
p
```



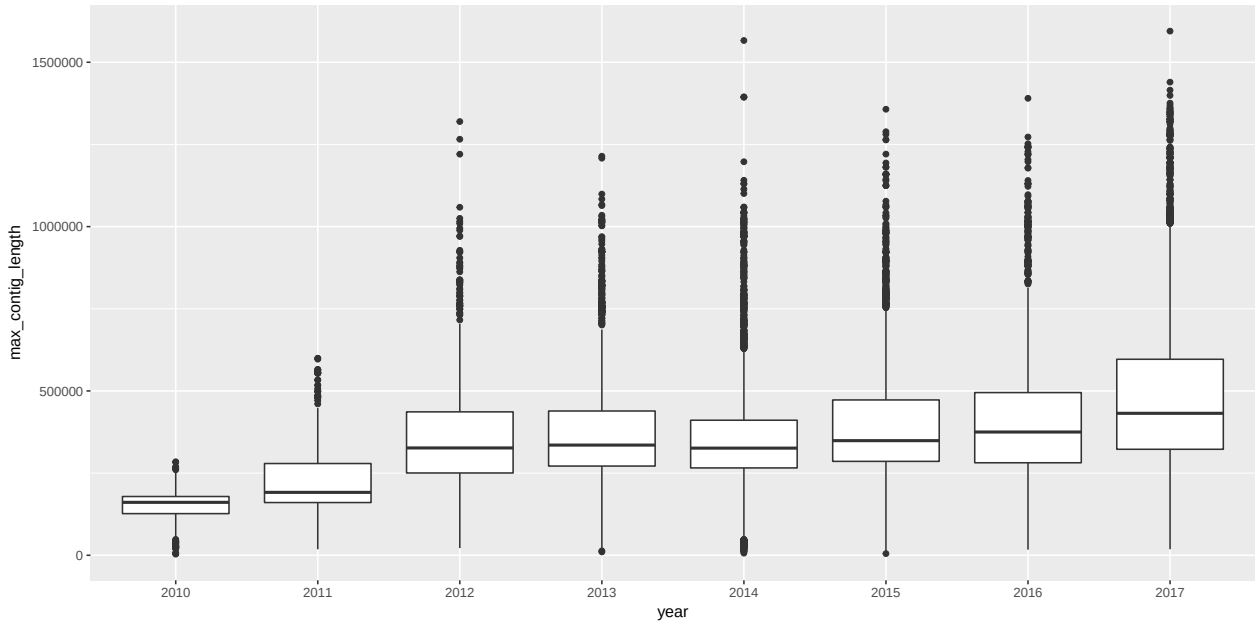
Mean Contig Length

```
p <- ggplot(metrics, aes(x = year, y = mean_contig_length)) +
    geom_boxplot()
p
```



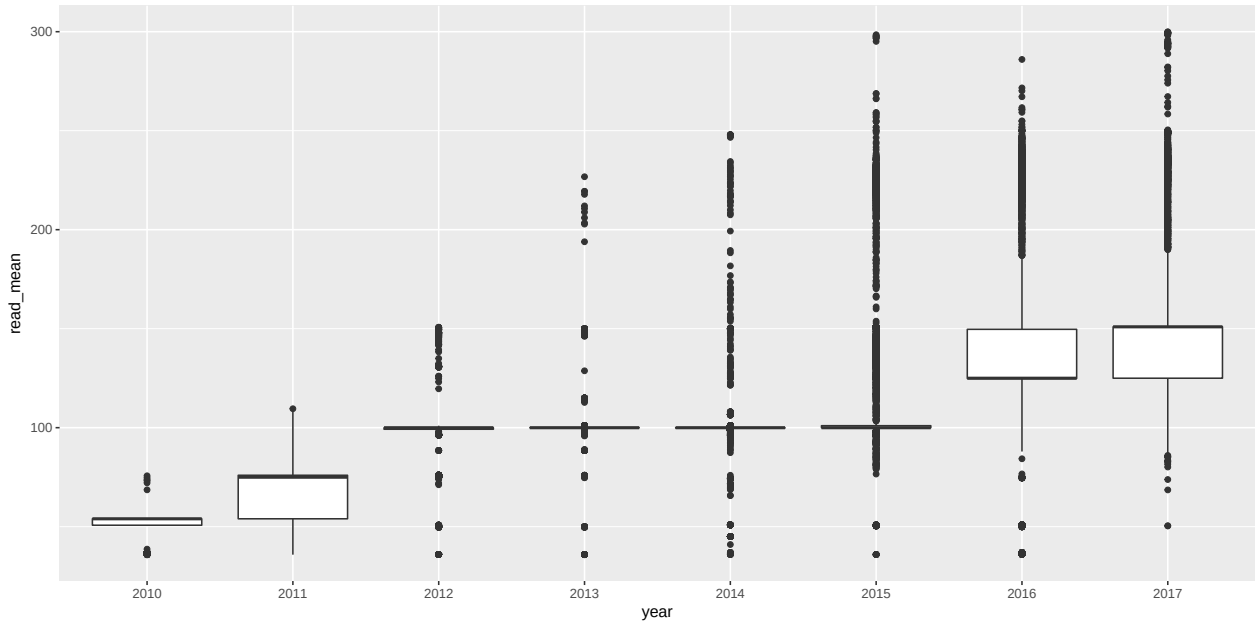
Max Contig Length

```
p <- ggplot(metrics, aes(x = year, y = max_contig_length)) +
    geom_boxplot()
p
```



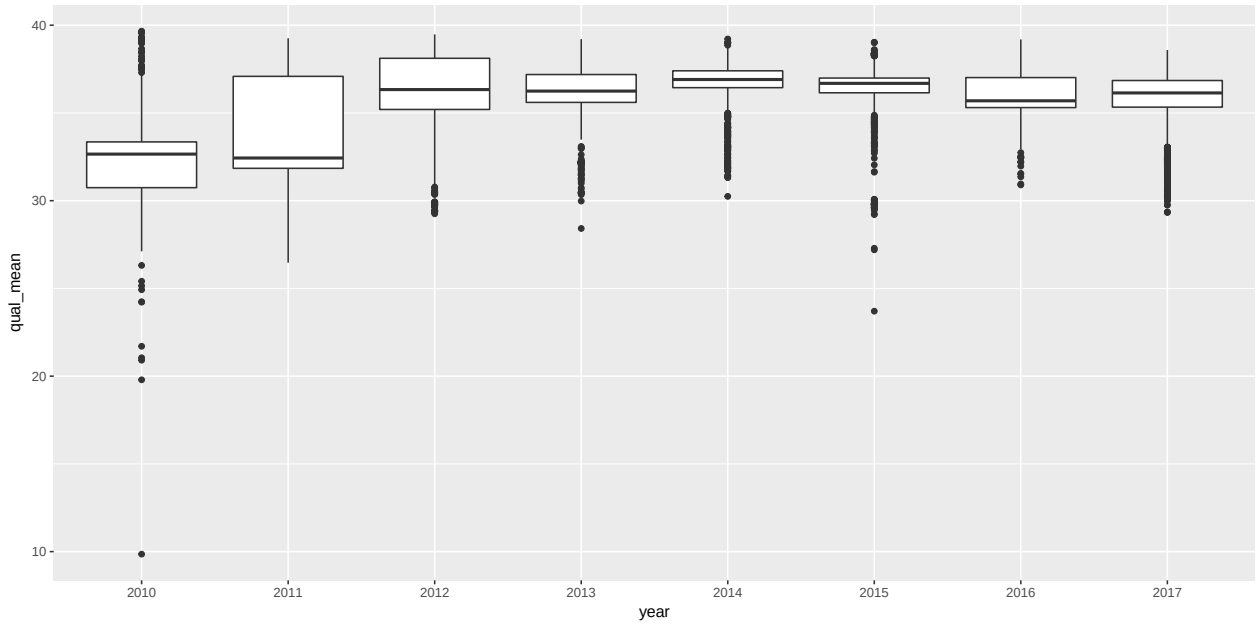
Mean Read Length

```
p <- ggplot(metrics, aes(x = year, y = read_mean)) +
    geom_boxplot()
p
```



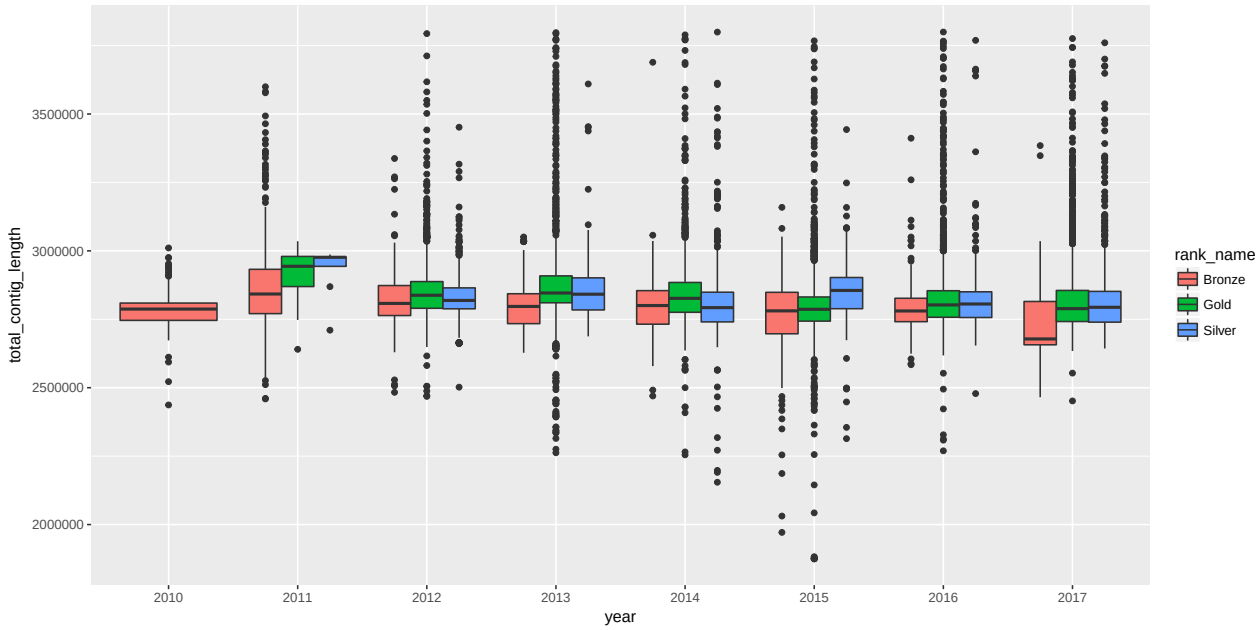
Mean Per-Read Quality Score

```
p <- ggplot(metrics, aes(x = year, y = qual_mean)) +
    geom_boxplot()
p
```



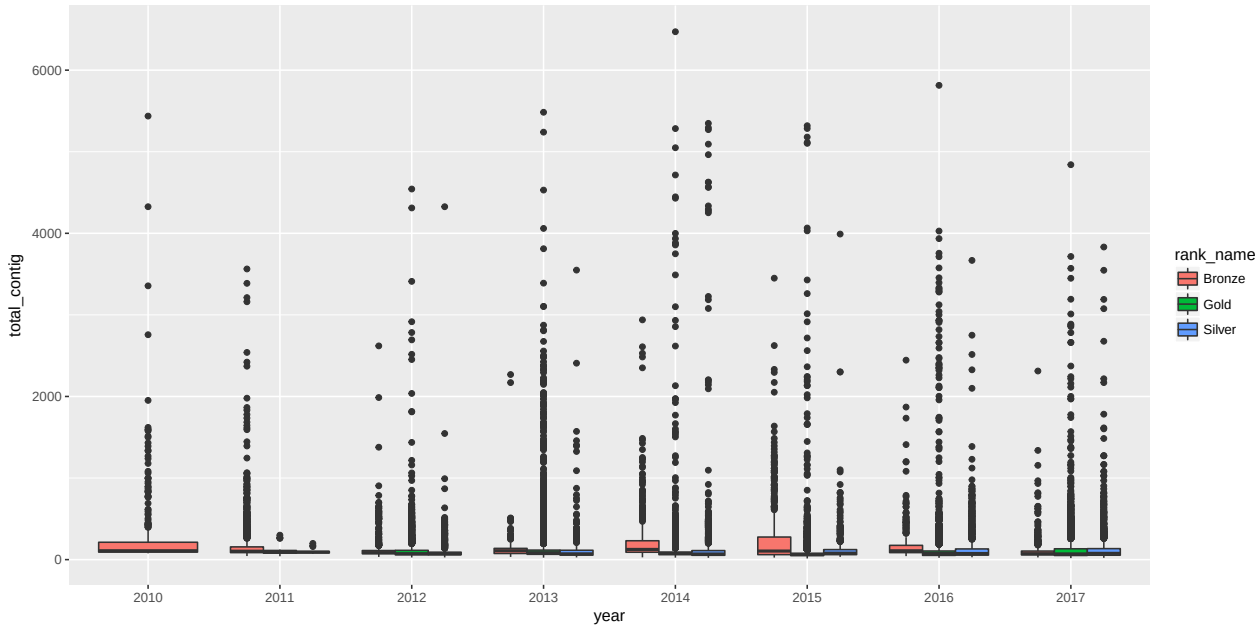
Assembly Size Grouped By Rank

```
p <- ggplot(metrics, aes(x = year, y = total_contig_length,
                         fill=rank_name, label=rank_name)) +
    geom_boxplot()
p
```



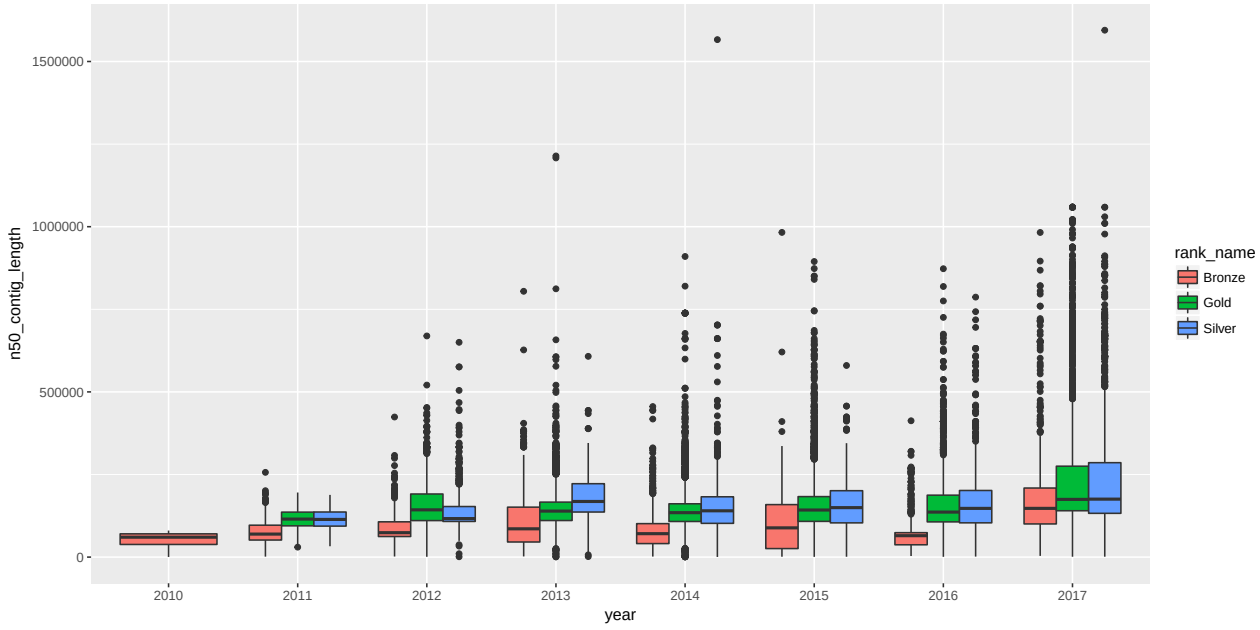
Total Contigs Grouped By Rank

```
p <- ggplot(metrics, aes(x = year, y = total_contig,
                         fill=rank_name, label=rank_name)) +
    geom_boxplot()
p
```



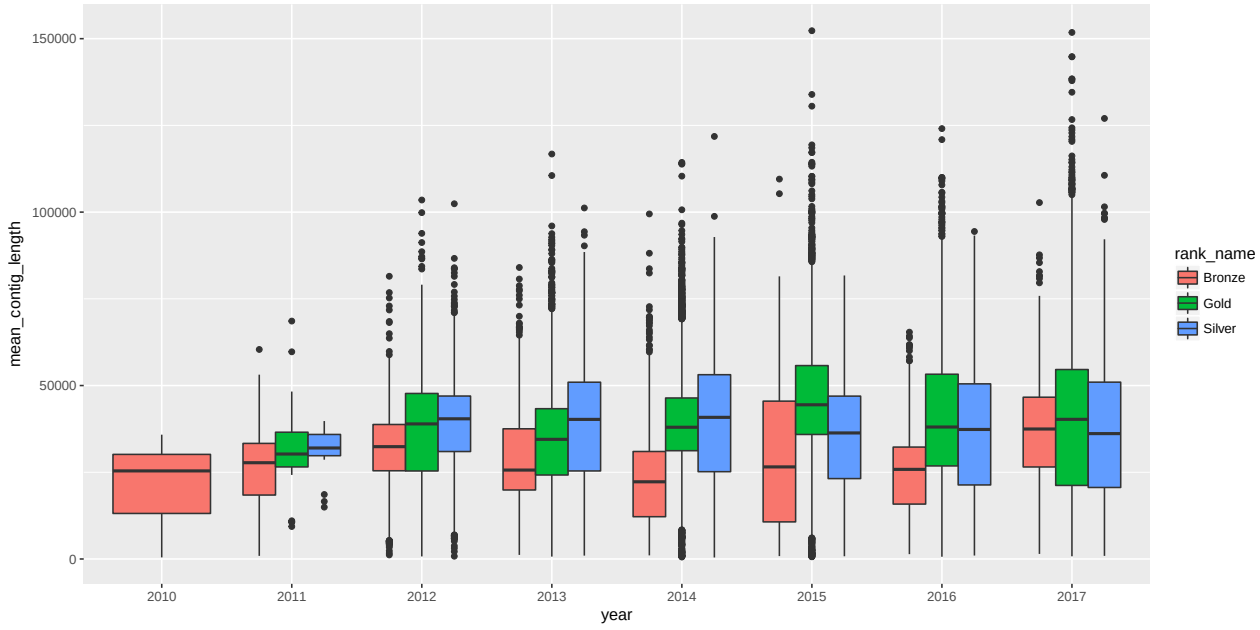
N50 Grouped By Rank

```
p <- ggplot(metrics, aes(x = year, y = n50_contig_length,
                         fill=rank_name, label=rank_name)) +
    geom_boxplot()
p
```



Mean Contig Length Grouped By Rank

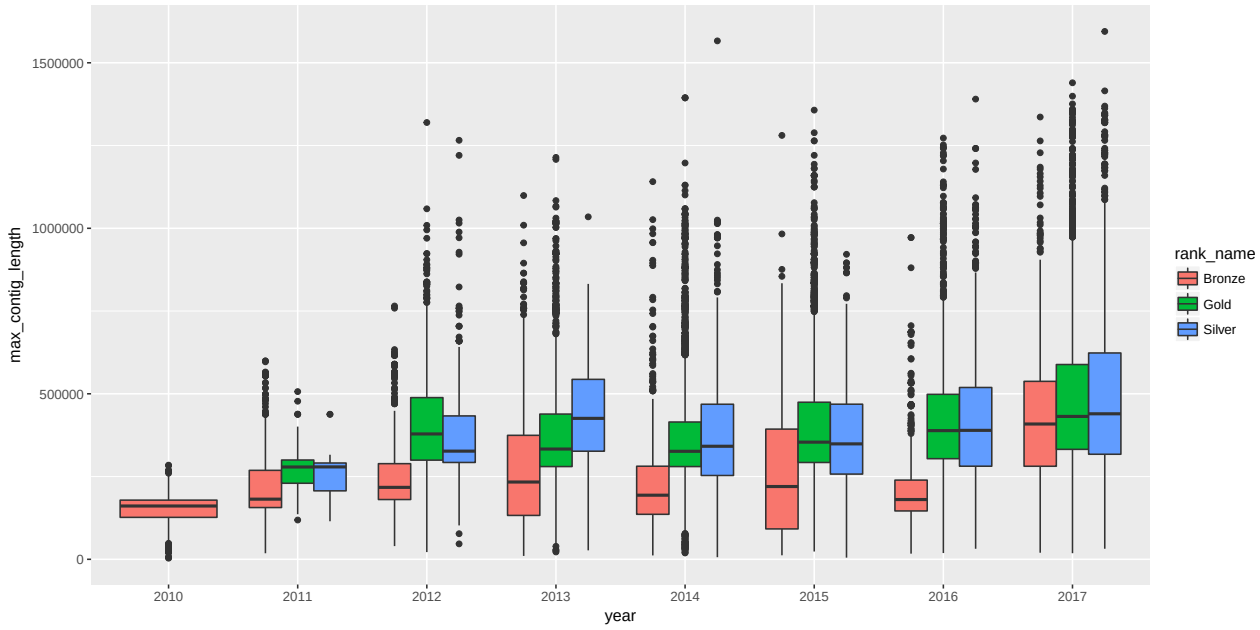
```
p <- ggplot(metrics, aes(x = year, y = mean_contig_length,
                         fill=rank_name, label=rank_name)) +
    geom_boxplot()
p
```



Max Contig Length Grouped By Rank

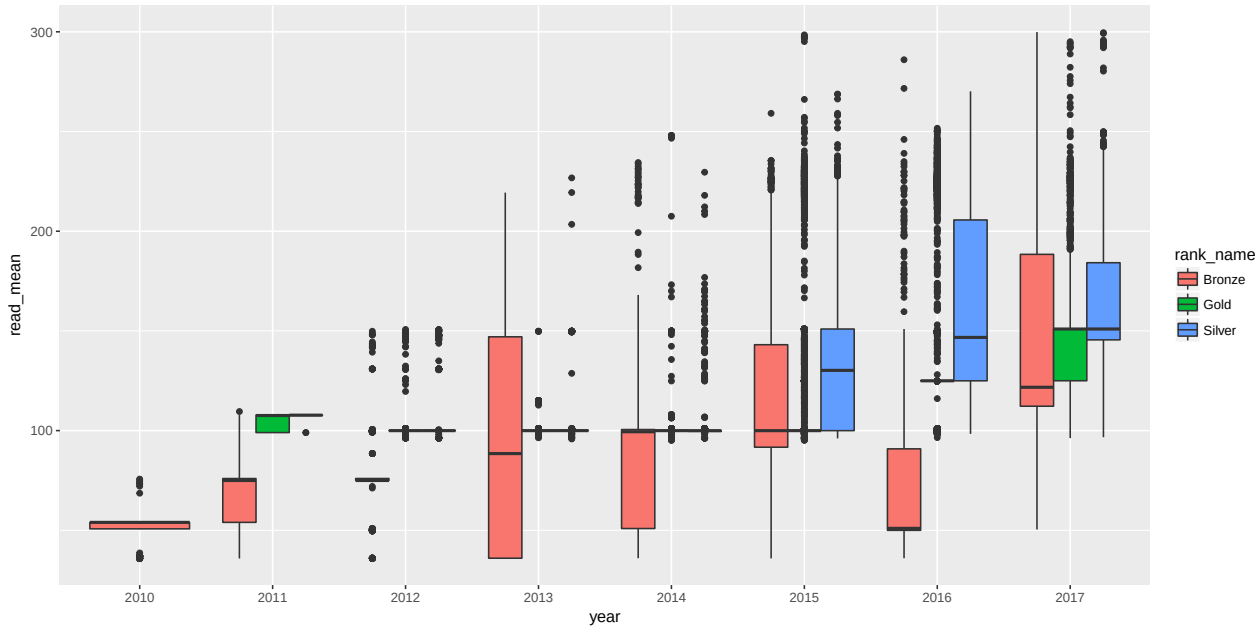
```
p <- ggplot(metrics, aes(x = year, y = max_contig_length,
                         fill=rank_name, label=rank_name)) +
```

```
    geom_boxplot()
p
```



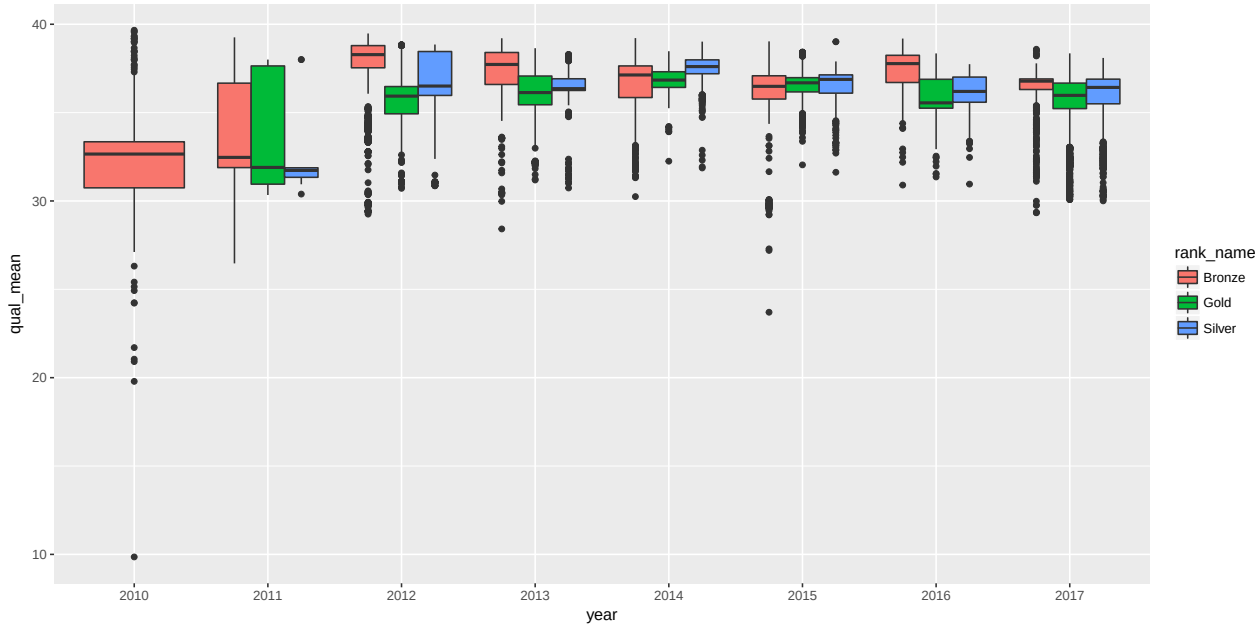
Mean Read Length Grouped By Rank

```
p <- ggplot(metrics, aes(x = year, y = read_mean,
                         fill=rank_name, label=rank_name)) +
    geom_boxplot()
p
```



Mean Per-Read Quality Score Grouped By Rank

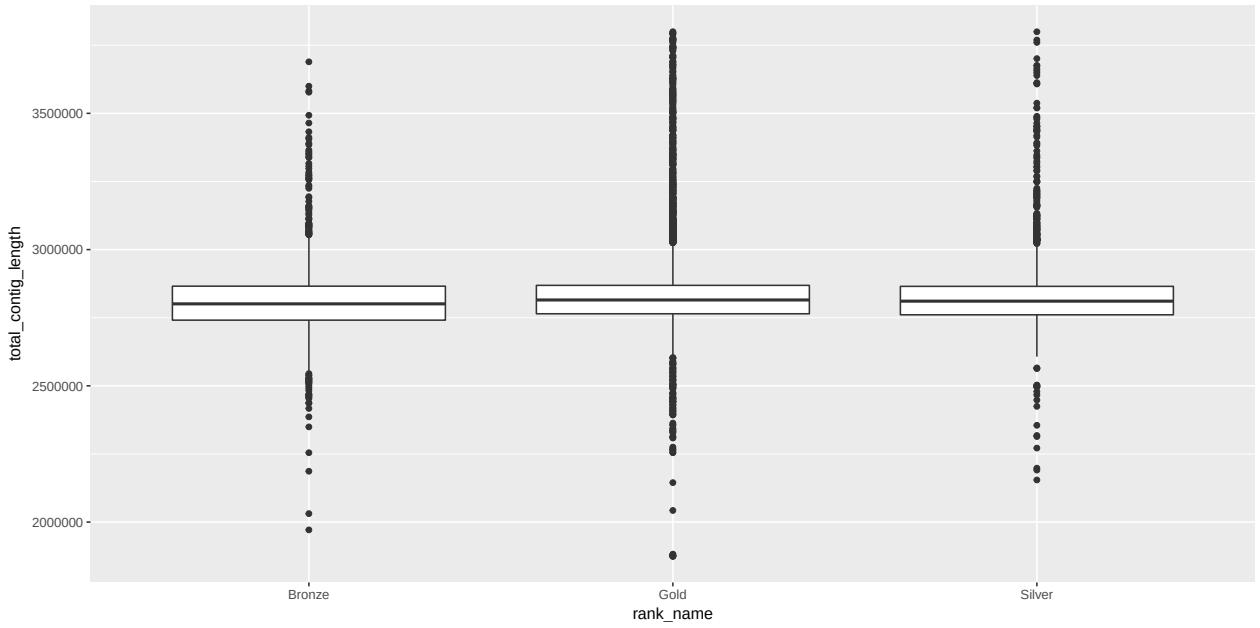
```
p <- ggplot(metrics, aes(x = year, y = qual_mean,
                         fill=rank_name, label=rank_name)) +
    geom_boxplot()
p
```



By Rank Plots

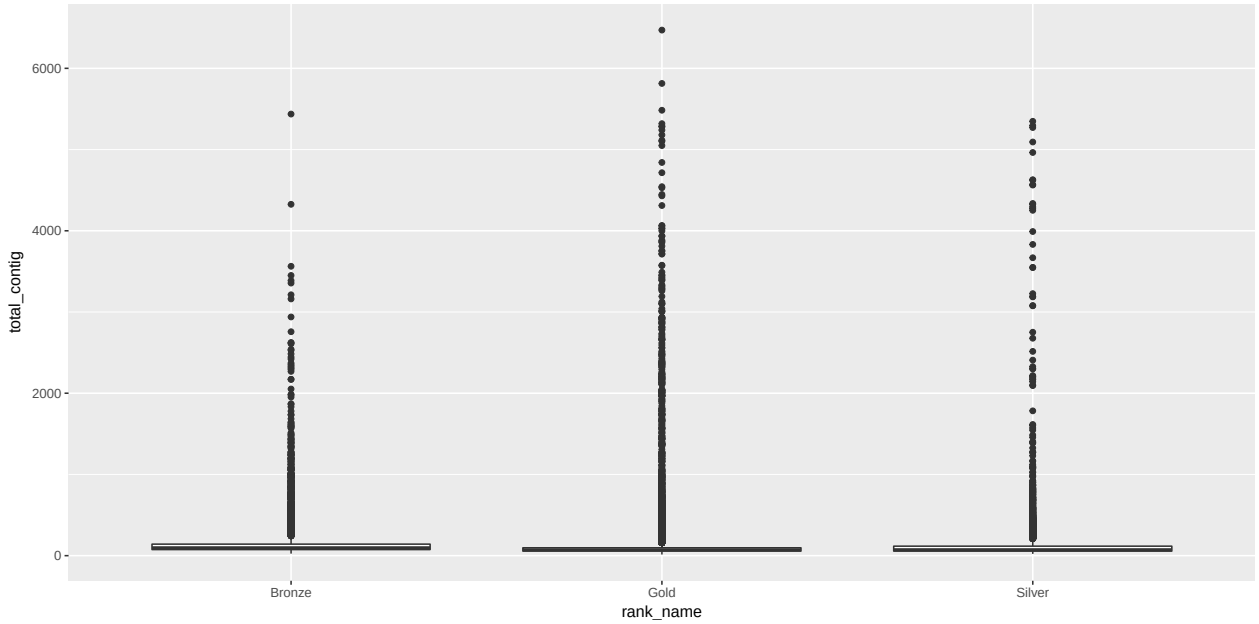
Assembly Size

```
p <- ggplot(metrics, aes(x = rank_name, y = total_contig_length)) +
    geom_boxplot()
p
```



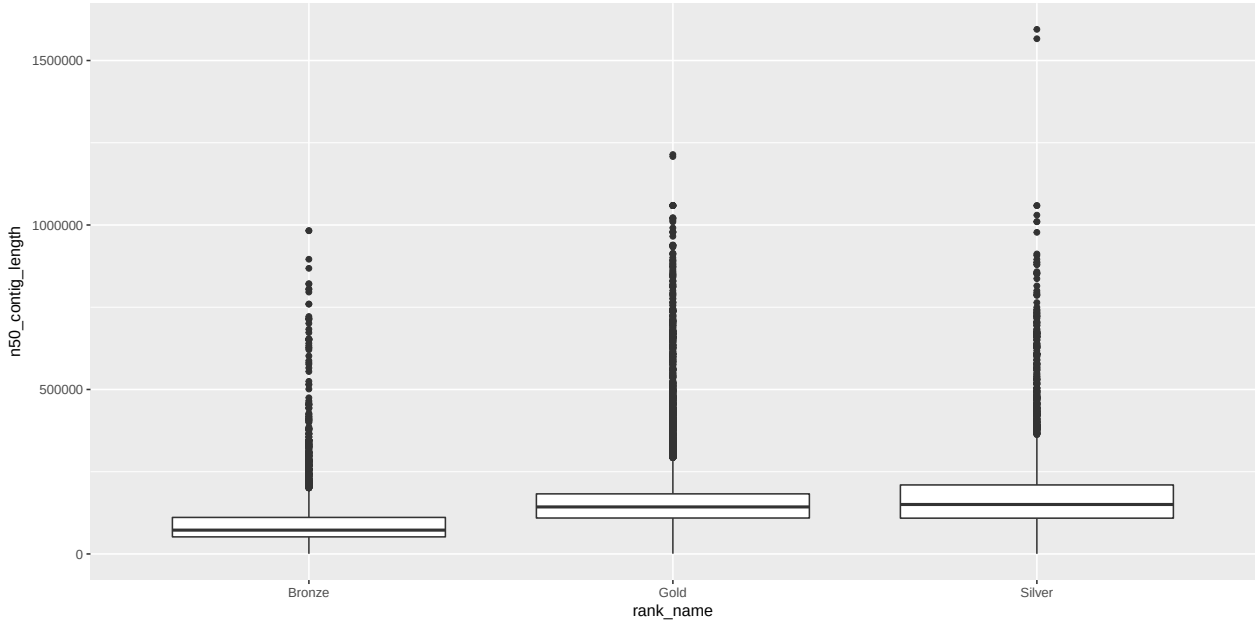
Total Contigs (smaller is better)

```
p <- ggplot(metrics, aes(x = rank_name, y = total_contig)) +
    geom_boxplot()
p
```



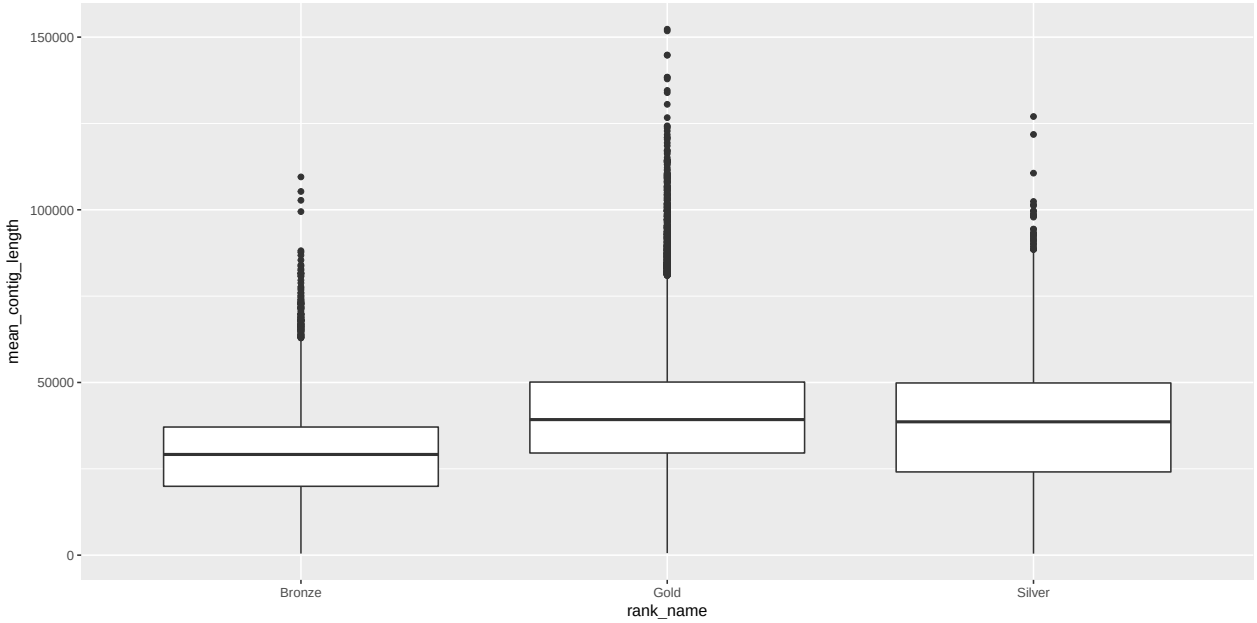
N50

```
p <- ggplot(metrics, aes(x = rank_name, y = n50_contig_length)) +
    geom_boxplot()
p
```



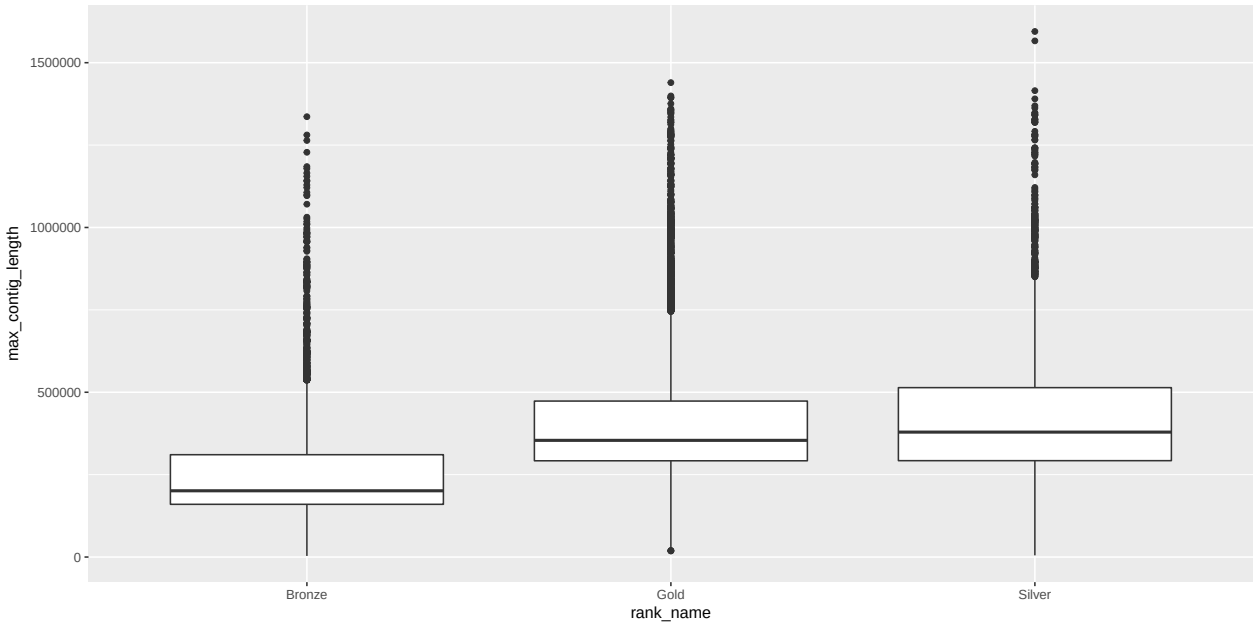
Mean Contig Length

```
p <- ggplot(metrics, aes(x = rank_name, y = mean_contig_length)) +
    geom_boxplot()
p
```



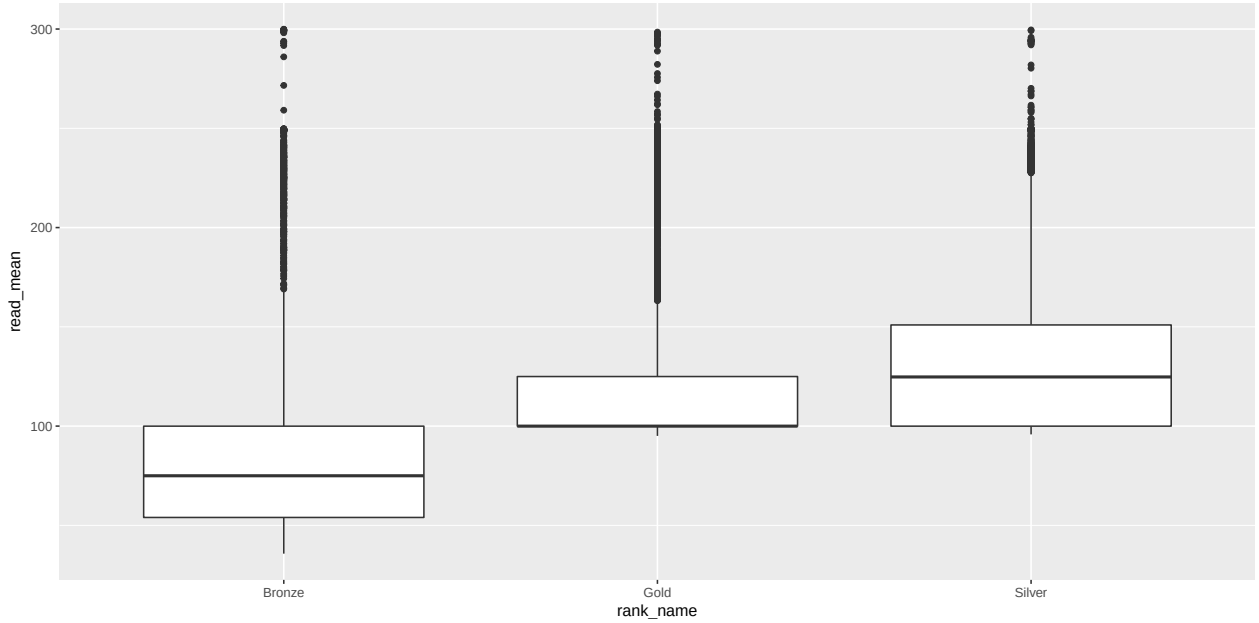
Max Contig Length

```
p <- ggplot(metrics, aes(x = rank_name, y = max_contig_length)) +
    geom_boxplot()
p
```



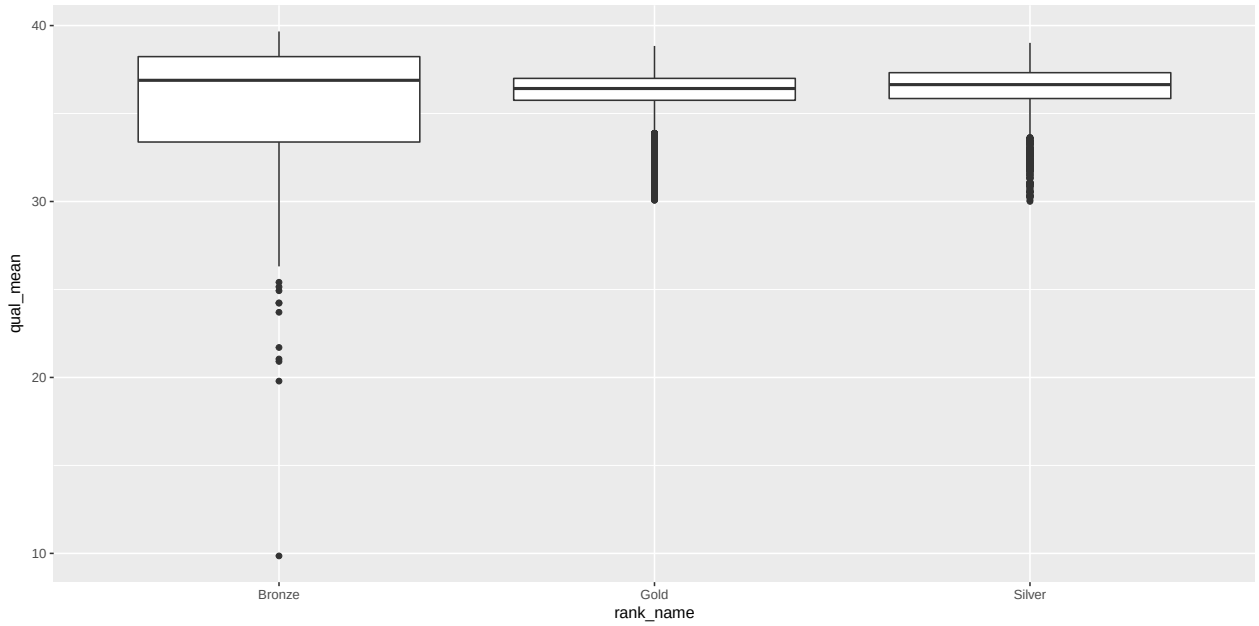
Mean Read Length

```
p <- ggplot(metrics, aes(x = rank_name, y = read_mean)) +
    geom_boxplot()
p
```



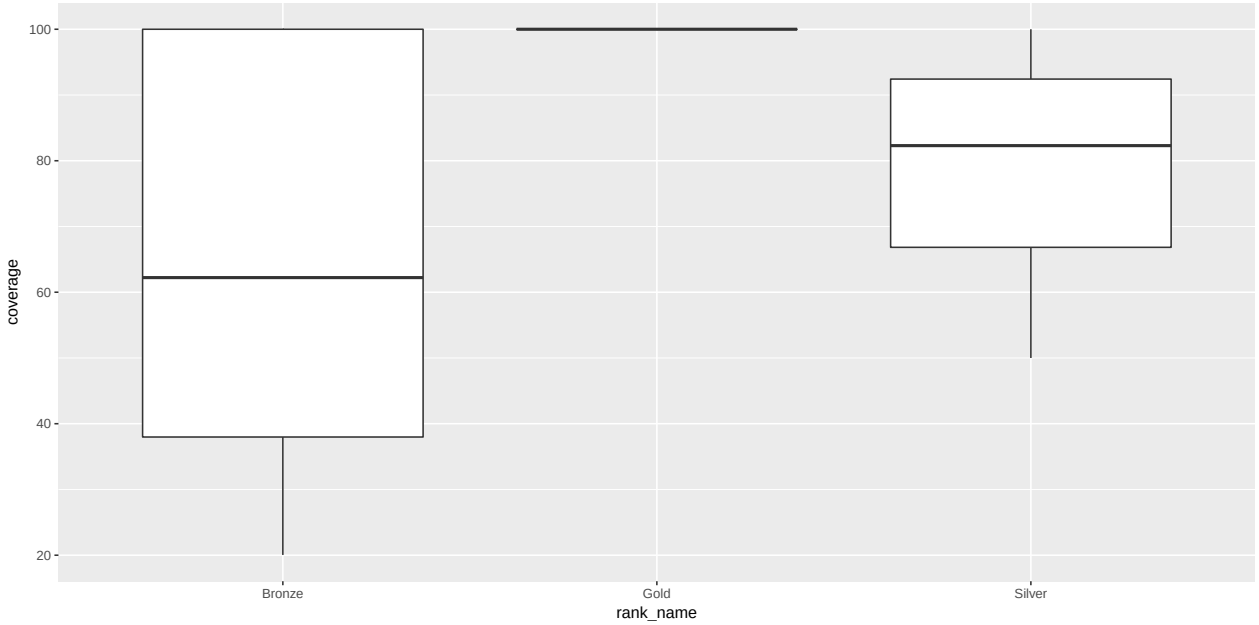
Mean Per-Read Quality Score

```
p <- ggplot(metrics, aes(x = rank_name, y = qual_mean)) +
    geom_boxplot()
p
```



Coverage

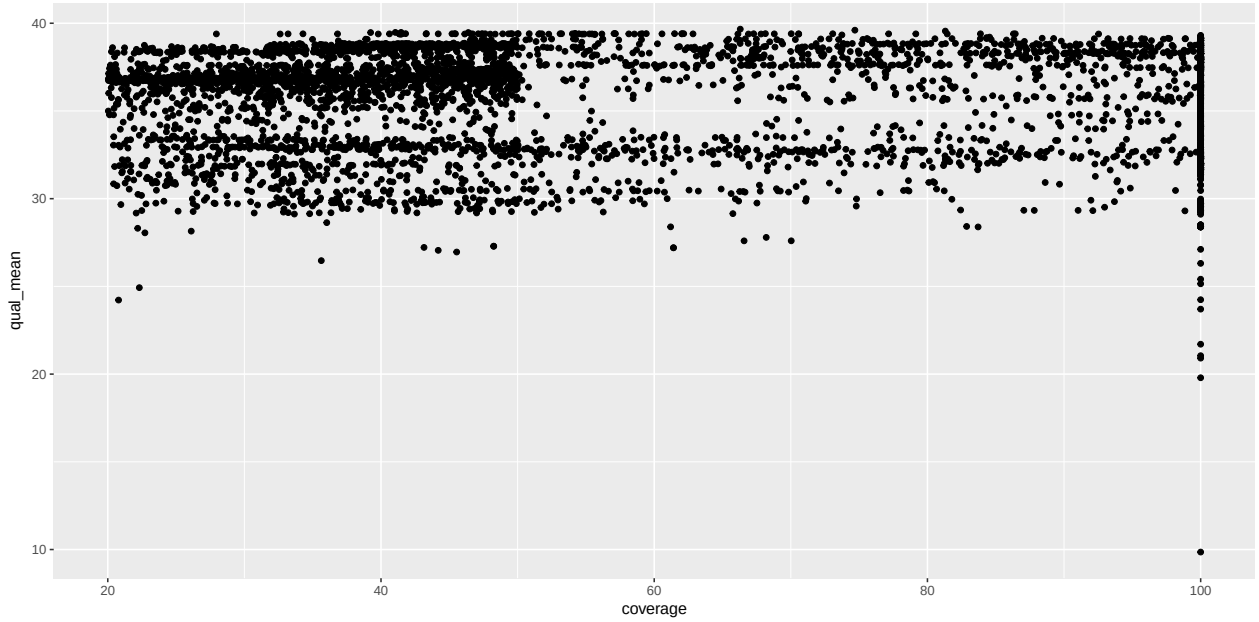
```
p <- ggplot(metrics, aes(x = rank_name, y = coverage)) +
    geom_boxplot()
p
```



Bronze Data

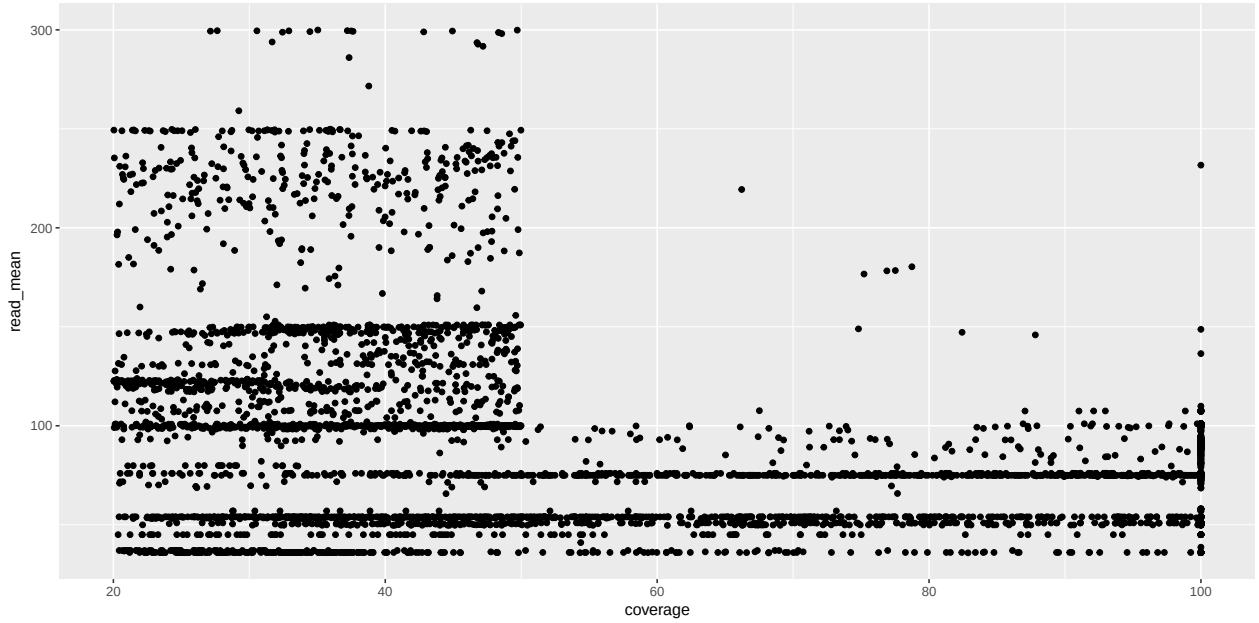
Coverage By Quality

```
p <- ggplot(metrics[metrics$rank.x == 1,], aes(x = coverage, y = qual_mean)) +
    geom_point()
p
```



Coverage By Read Length

```
p <- ggplot(metrics[metrics$rank.x == 1,], aes(x = coverage, y = read_mean)) +
    geom_point()
p
```



Session Info

```
sessionInfo()
```

```
## R version 3.4.3 (2017-11-30)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.2 LTS
##
## Matrix products: default
## BLAS: /usr/lib/libblas/libblas.so.3.6.0
## LAPACK: /usr/lib/lapack/liblapack.so.3.6.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] reshape2_1.4.3  ggplot2_2.2.1   staphopia_0.1.9
##
## loaded via a namespace (and not attached):
```

```
##  [1] Rcpp_0.12.15        knitr_1.20          magrittr_1.5
##  [4] munsell_0.4.3       colorspace_1.3-2    R6_2.2.2
##  [7] rlang_0.1.6         stringr_1.2.0       httr_1.3.1
## [10] plyr_1.8.4          tools_3.4.3         grid_3.4.3
## [13] data.table_1.10.4-3 gtable_0.2.0        htmltools_0.3.6
## [16] yaml_2.1.18         lazyeval_0.2.1      rprojroot_1.3-2
## [19] digest_0.6.15       tibble_1.4.2        curl_3.1
## [22] evaluate_0.10.1     rmarkdown_1.9       labeling_0.3
## [25] stringi_1.1.6       compiler_3.4.3      pillar_1.1.0
## [28] scales_0.5.0        backports_1.1.2     jsonlite_1.5
```