Results Section: Public Sequencing Metrics

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

Aggregating Data For Public Samples

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

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

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

```r
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.

```r
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`.

```r
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'
    )
)
```

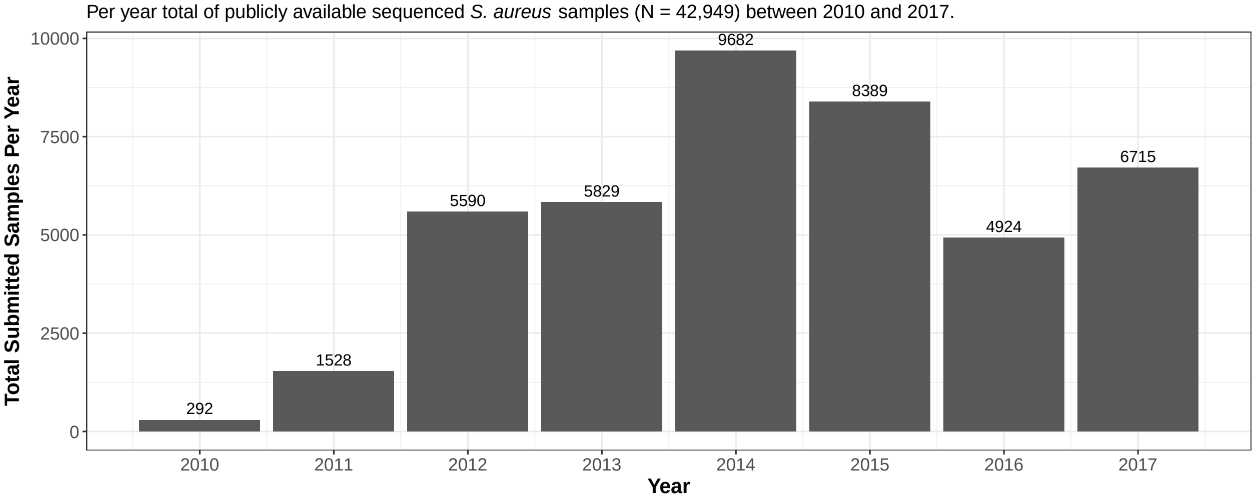
Visualizing Metrics

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

By Year Plots

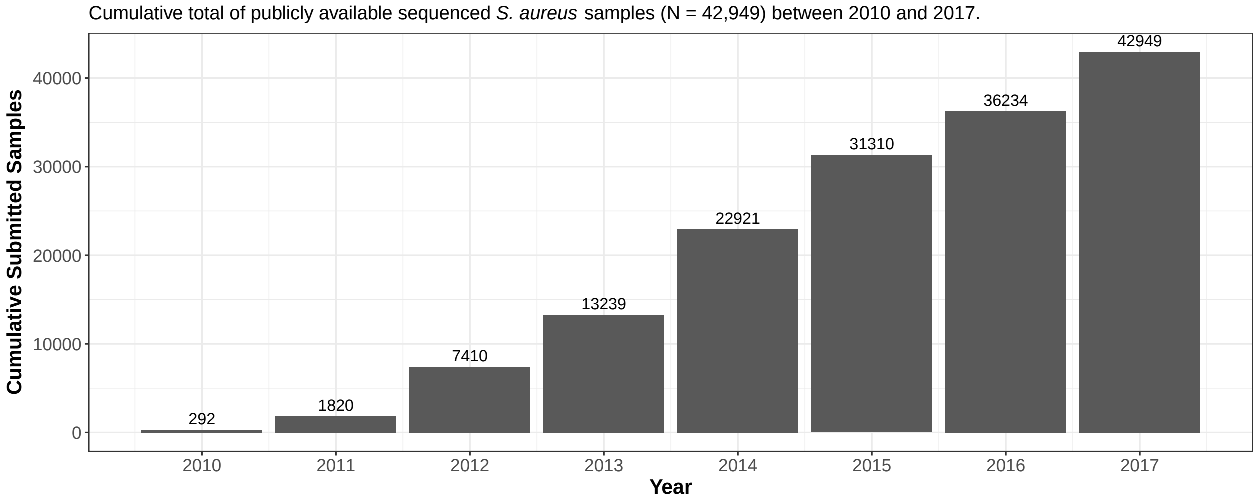
Submissions Per Year

```
title <- substitute(paste("Per year total of publicly available sequenced ",
                          italic('S. aureus')," samples (N = ", x,") between ", min_year, " and ", max_
                    list(x=format(max(submissions$overall), big.mark=',', scientific=FALSE),
                         min_year=min(submissions$year),
                         max_year=max(submissions$year)
))
p <- ggplot(data=submissions, aes(x=year, y=count)) +
    xlab("Year") +
    ylab("Total Submitted Samples Per Year") +
    ggtitle(title) +
    geom_bar(stat='identity') +
    geom_text(aes(label=count), vjust = -0.5) +
    scale_x_continuous(breaks = round(seq(min(submissions$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
```



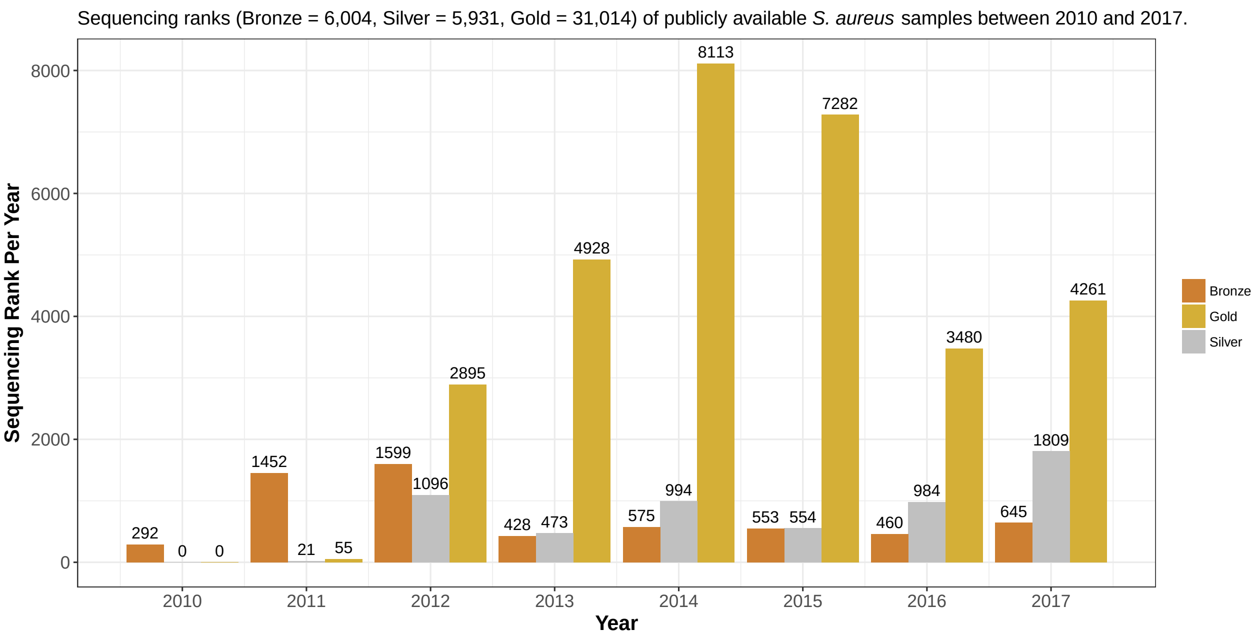
Overall Submissions

```
title <- substitute(paste("Cumulative total of publicly available sequenced ",
                          italic('S. aureus')," samples (N = ", x,") between ", min_year, " and ", max_
                    list(x=format(max(submissions$overall), big.mark=',', scientific=FALSE),
                         min_year=min(submissions$year),
                         max_year=max(submissions$year)
))
p <- ggplot(data=submissions, aes(x=year, y=overall)) +
    xlab("Year") +
    ylab("Cumulative Submitted Samples") +
    ggtitle(title) +
    geom_bar(stat='identity') +
    geom_text(aes(label=overall), vjust = -0.5) +
    scale_x_continuous(breaks = round(seq(min(submissions$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
```



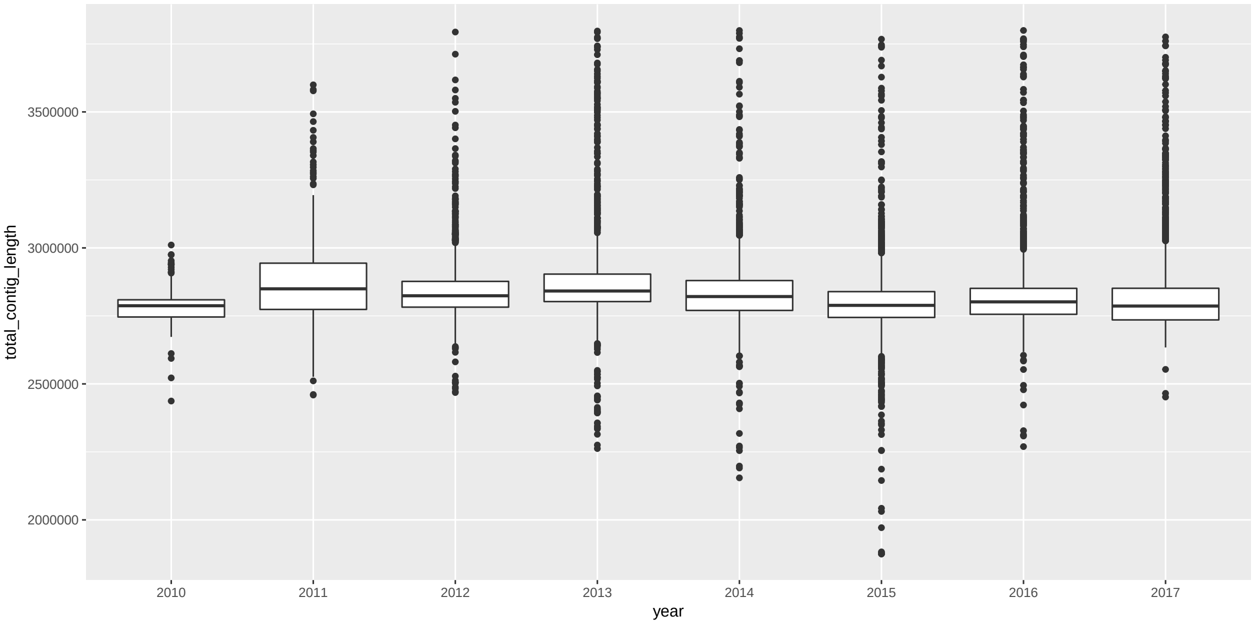
Submission Ranks

```
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))

title <- substitute(paste("Sequencing ranks (Bronze = ", b, ", Silver = ", s,
                          ", Gold = ", g, ") of publicly available ",
                          italic('S. aureus')," samples between ", min_year,
                          " and ", max_year, "."), list(
    b=format(max(ranks$overall_bronze), big.mark=',', scientific=FALSE),
    s=format(max(ranks$overall_silver), big.mark=',', scientific=FALSE),
    g=format(max(ranks$overall_gold), big.mark=',', scientific=FALSE),
    min_year=min(ranks$year),
    max_year=max(ranks$year)
))
p <- ggplot(data=melted, aes(x=year, y=value, fill=title, group=rank, label=title)) +
    xlab("Year") +
    ylab("Sequencing Rank Per Year") +
    ggtitle(title) +
    geom_bar(stat='identity', position='dodge') +
    geom_text(aes(label=value), vjust = -0.5, position = position_dodge(.9)) +
    scale_fill_manual(values=c("#CD7F32", "#D4AF37", "#C0C0C0")) +
    scale_x_continuous(breaks = round(seq(min(ranks$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
```



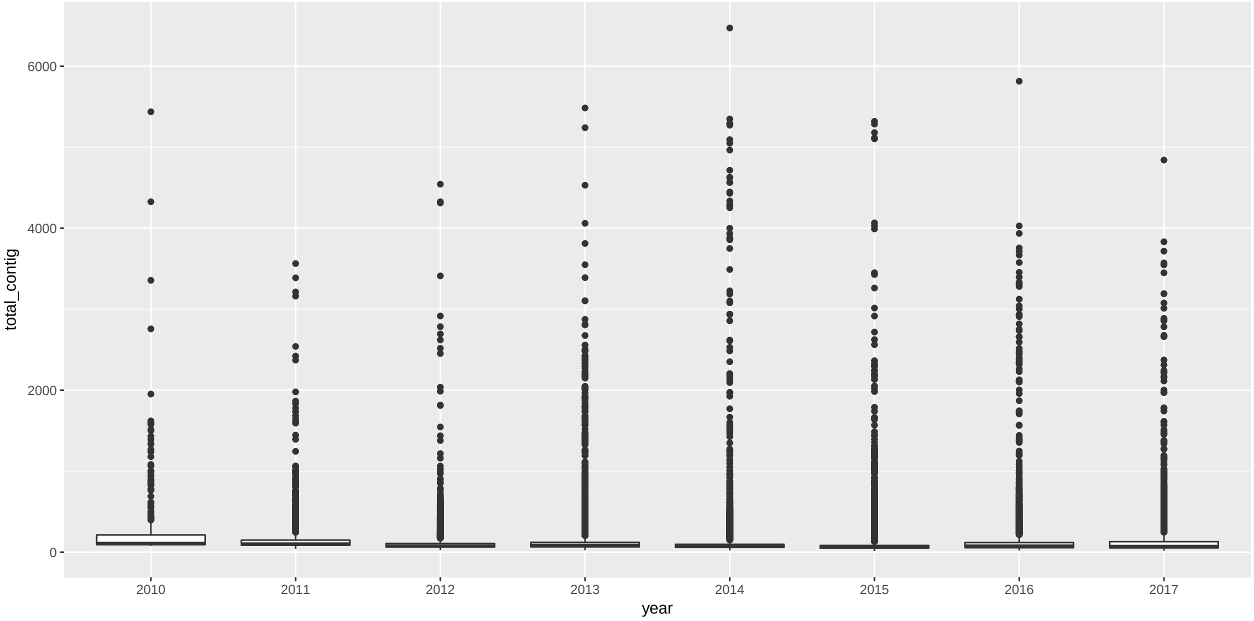
Assembly Size

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



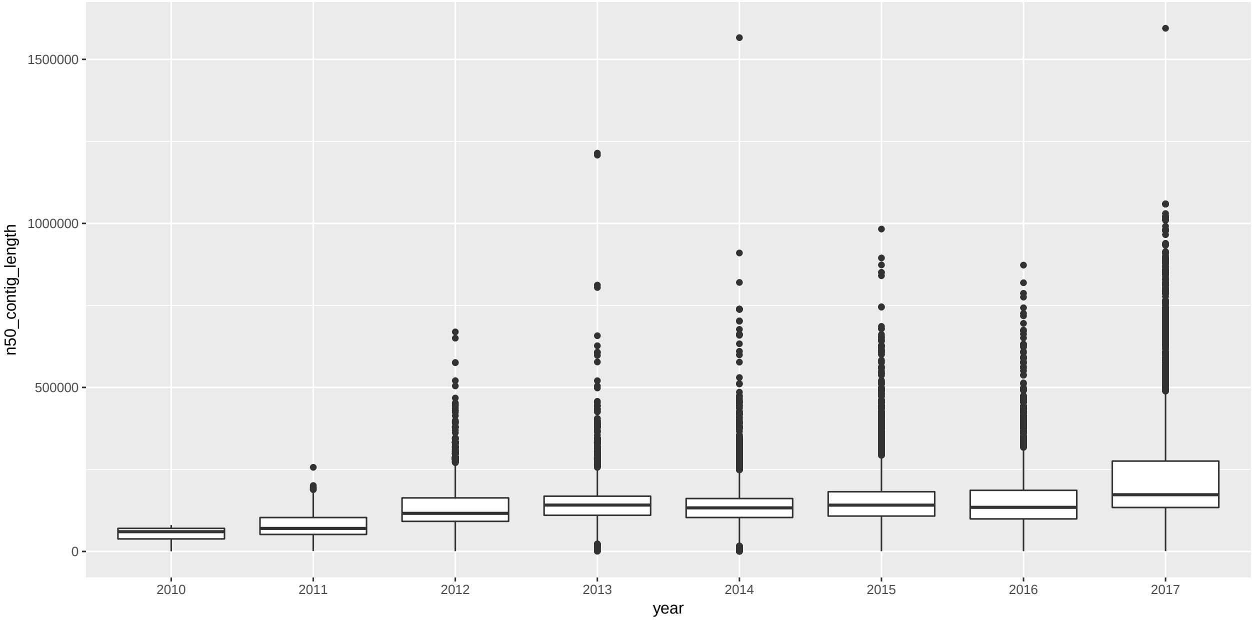
Total Contigs (smaller is better)

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



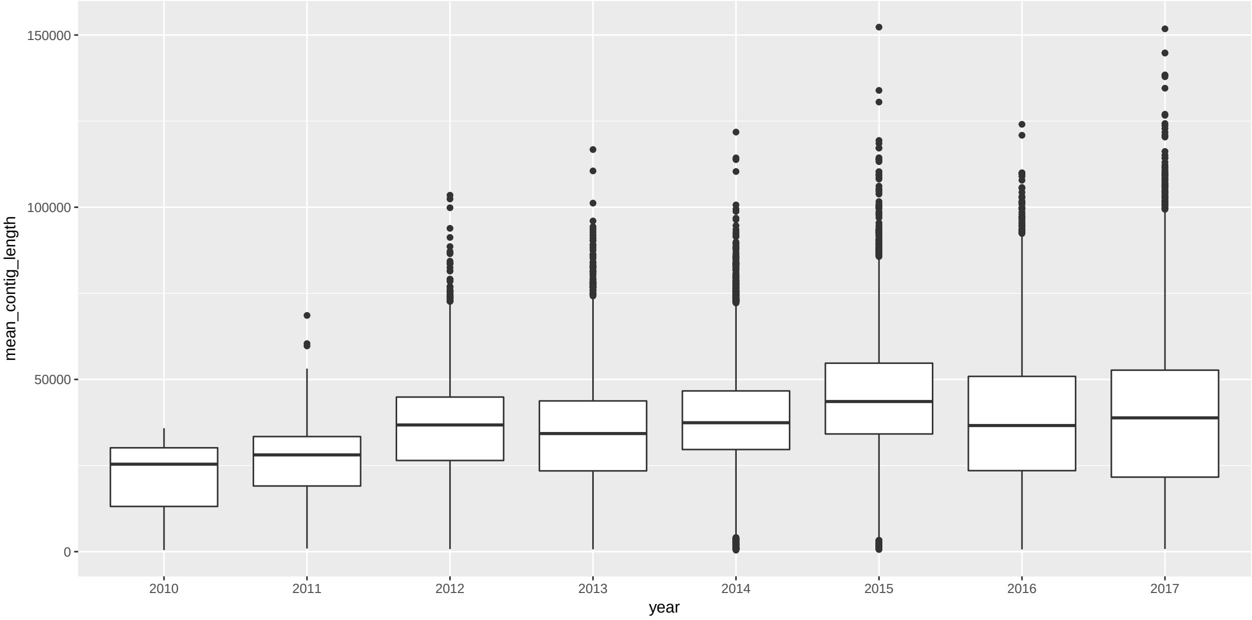
N50

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



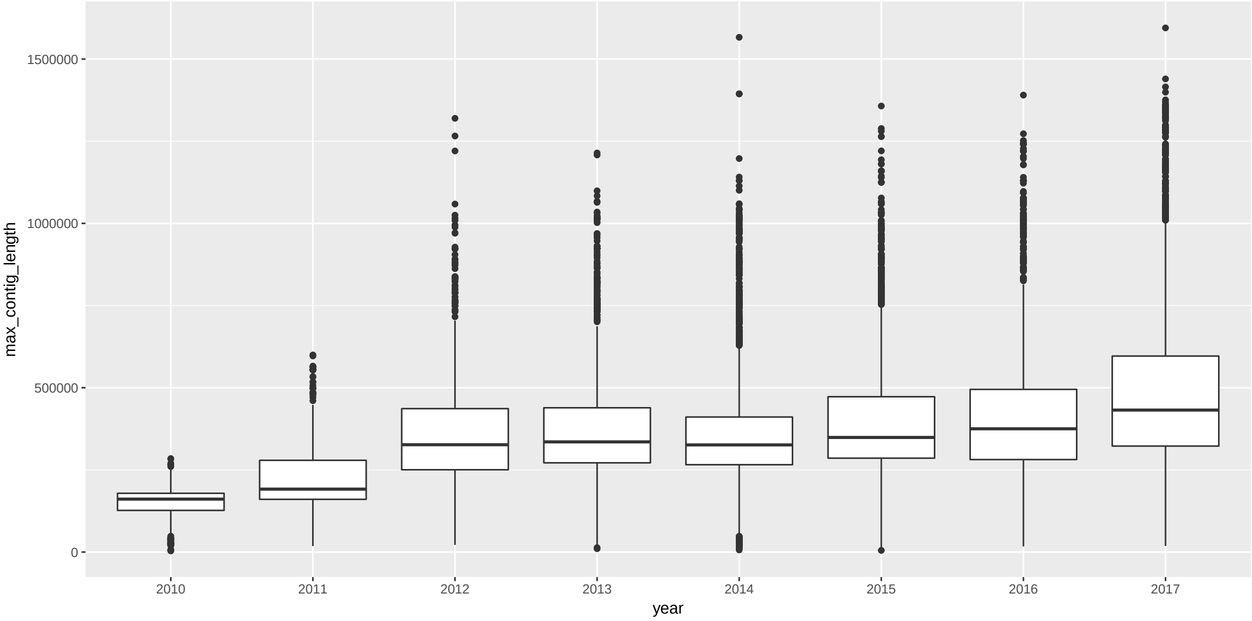
Mean Contig Length

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



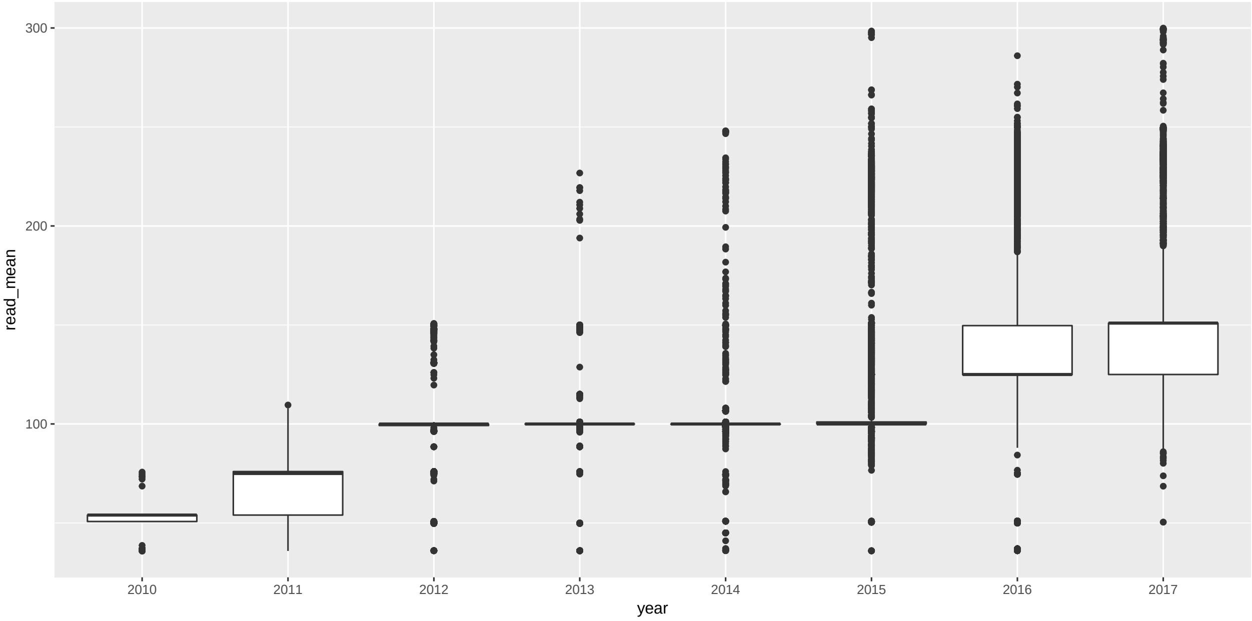
Max Contig Length

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



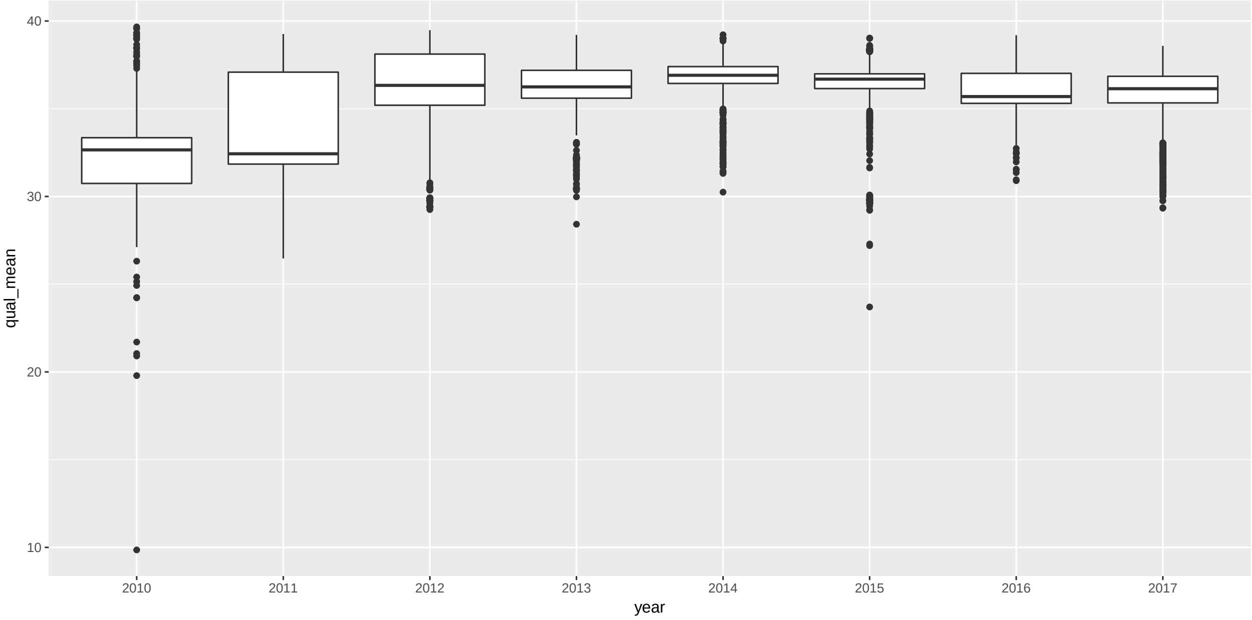
Mean Read Length

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



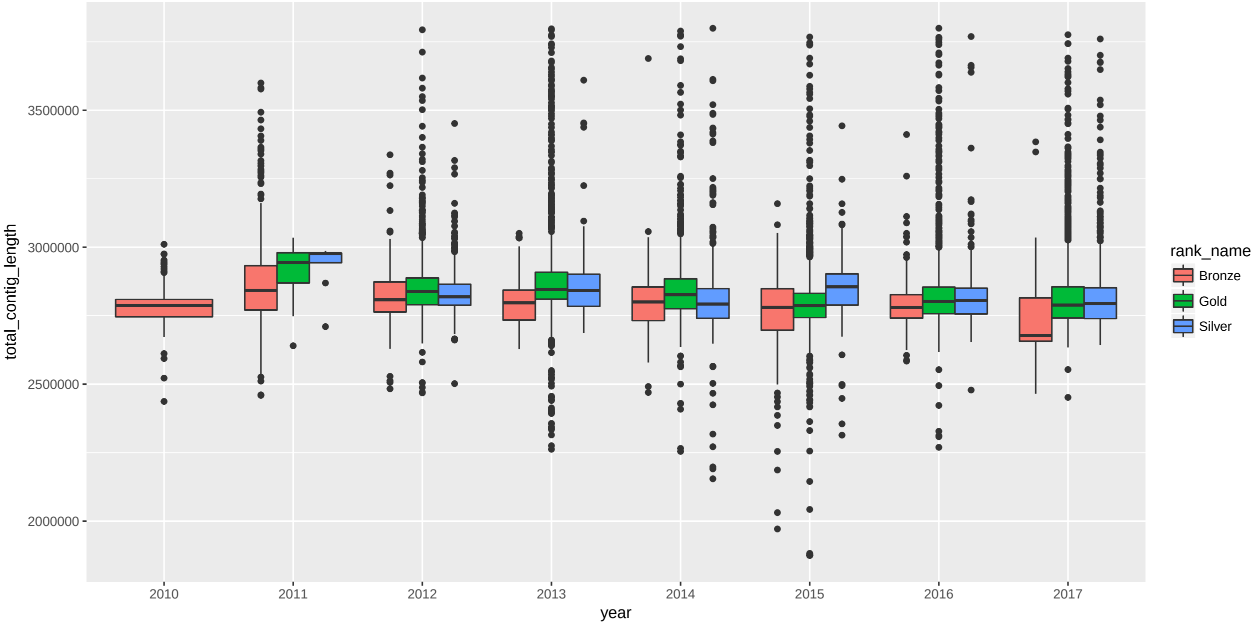
Mean Per-Read Quality Score

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



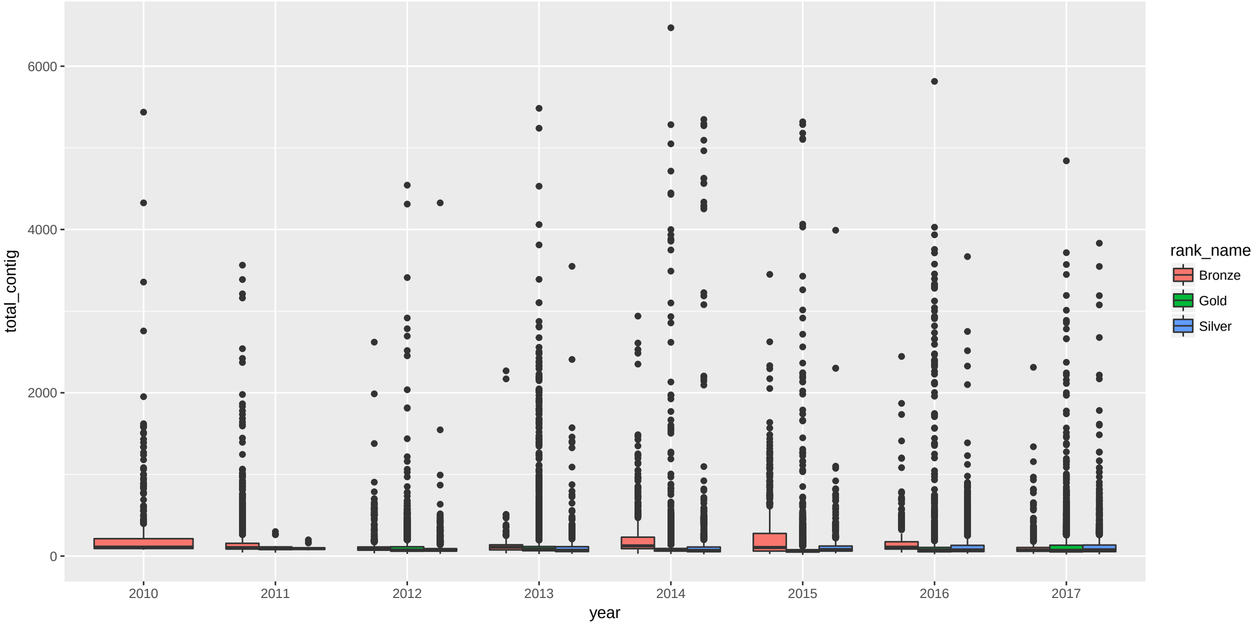
Assembly Size Grouped By Rank

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



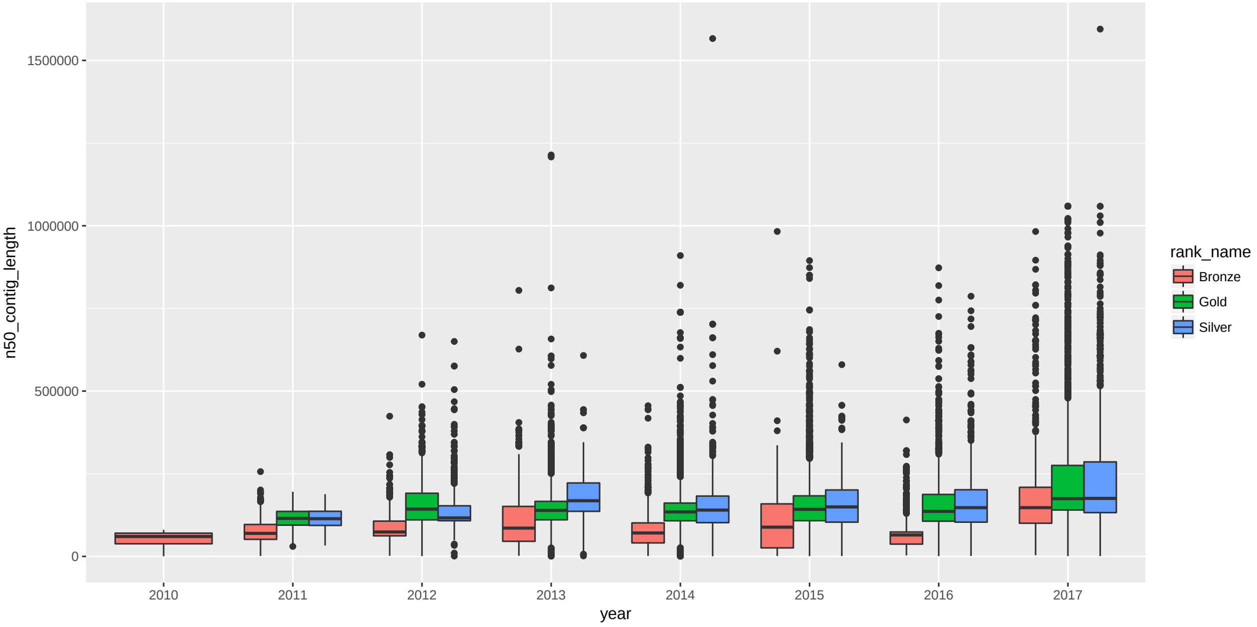
Total Contigs Grouped By Rank

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



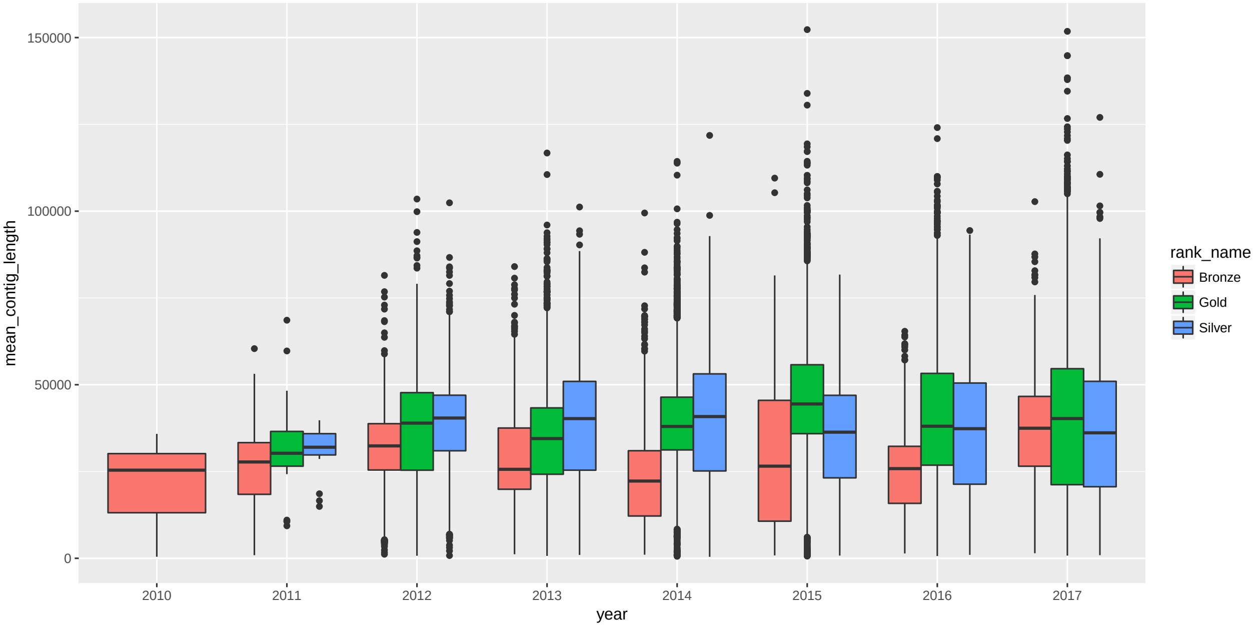
N50 Grouped By Rank

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



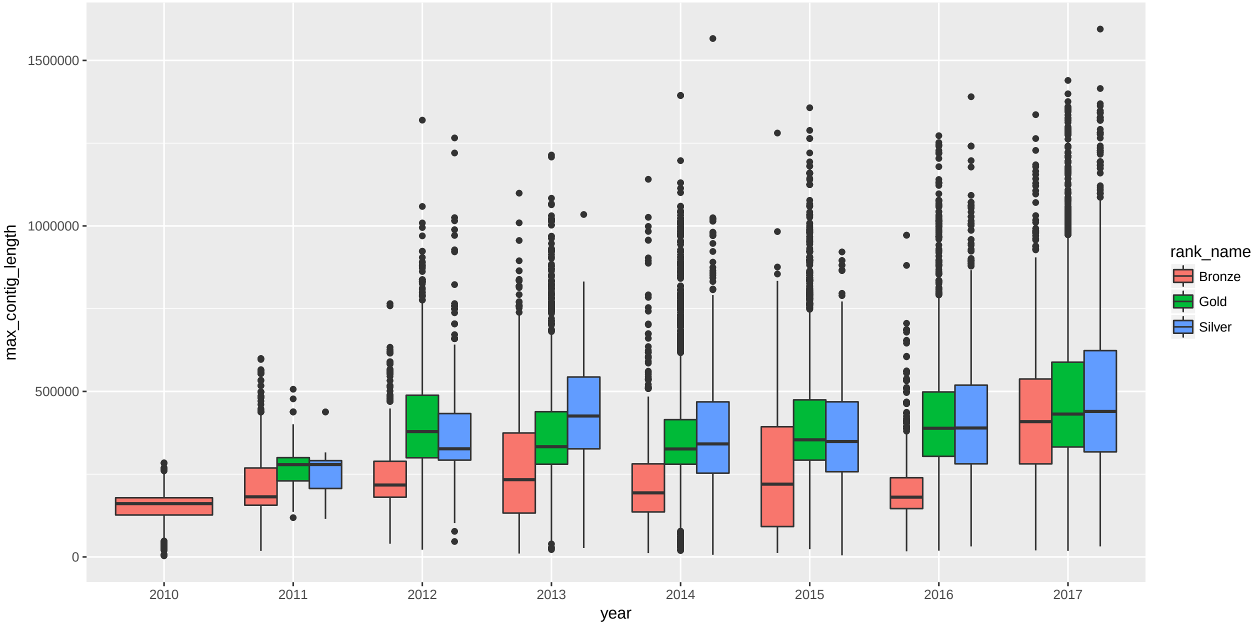
Mean Contig Length Grouped By Rank

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



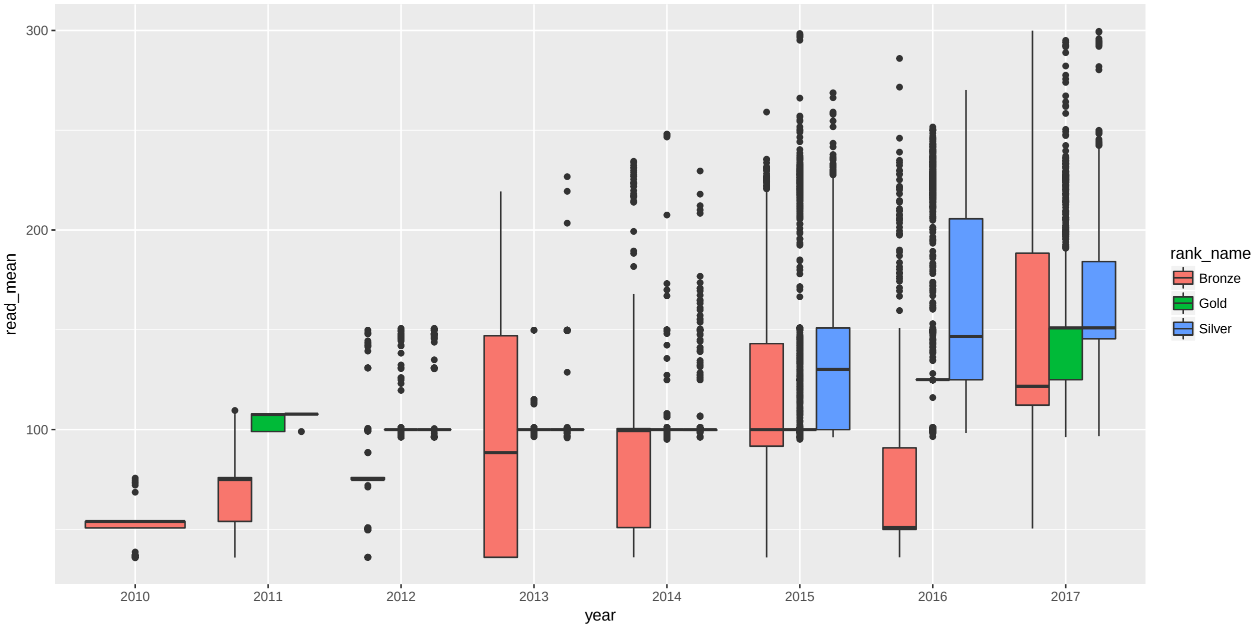
Max Contig Length Grouped By Rank

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



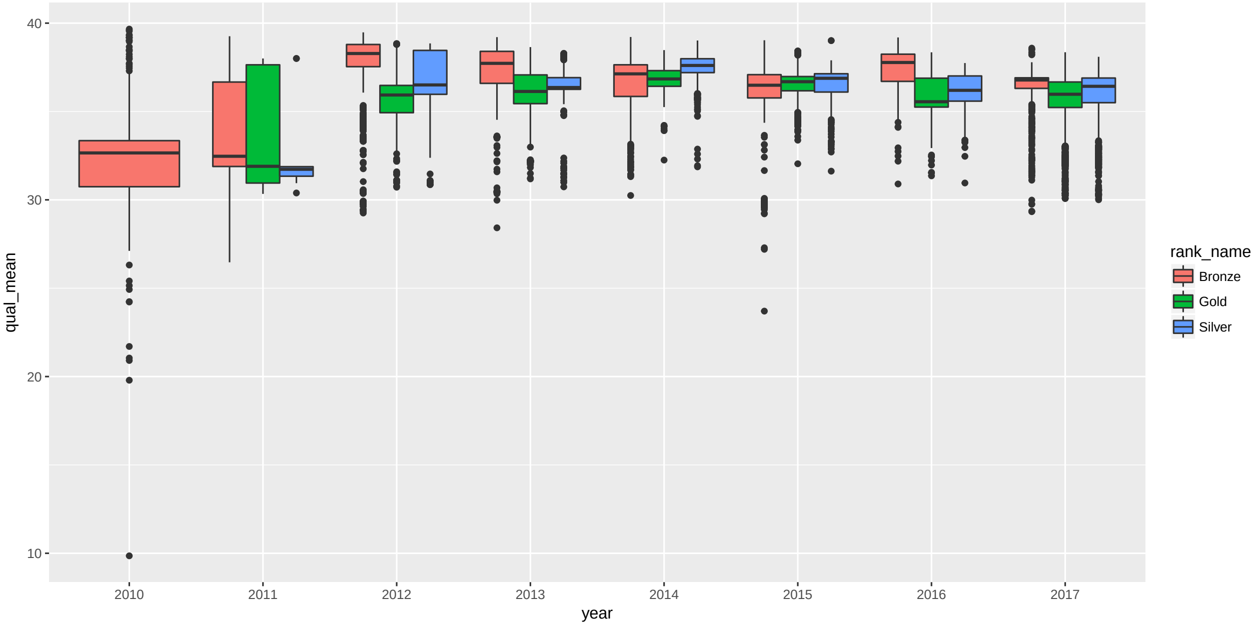
Mean Read Length Grouped By Rank

```
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

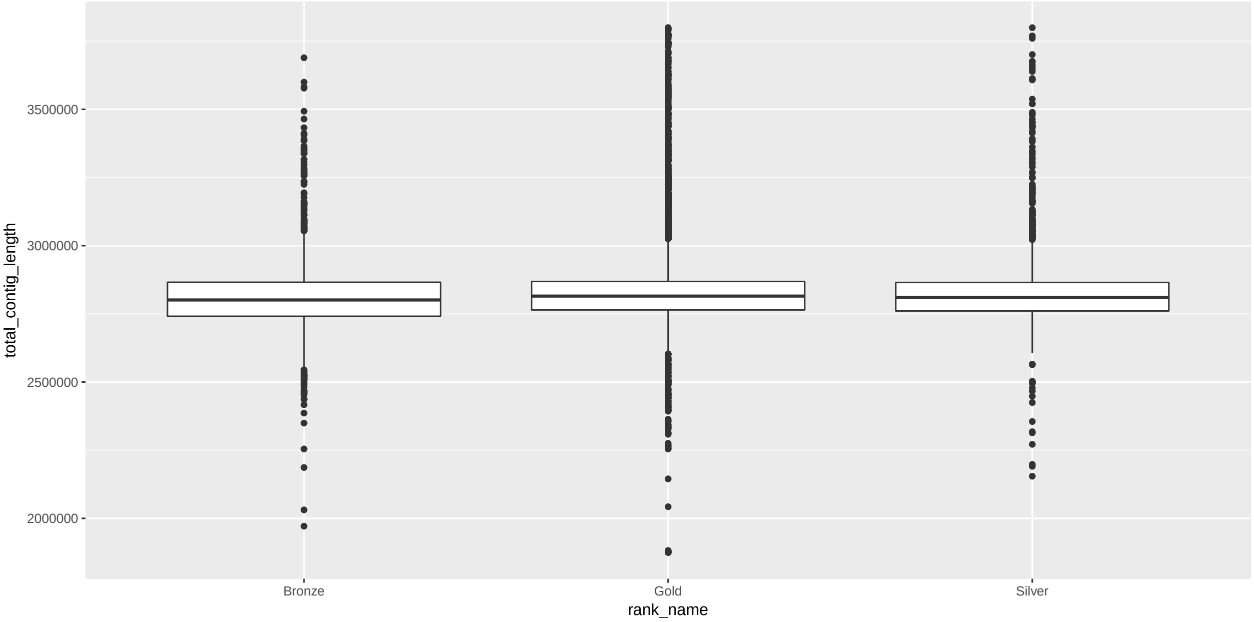
```
p <- ggplot(metrics, aes(x = year, y = qual_mean,
                         fill=rank_name, label=rank_name)) +
  geom_boxplot()
p
```



By Rank Plots

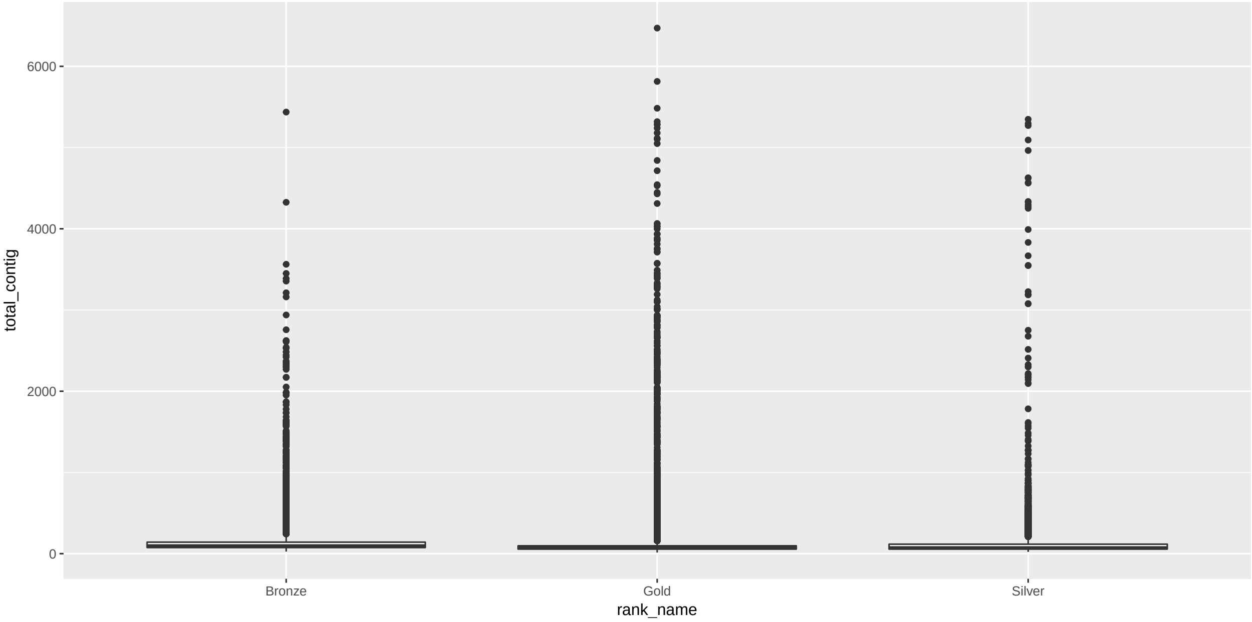
Assembly Size

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



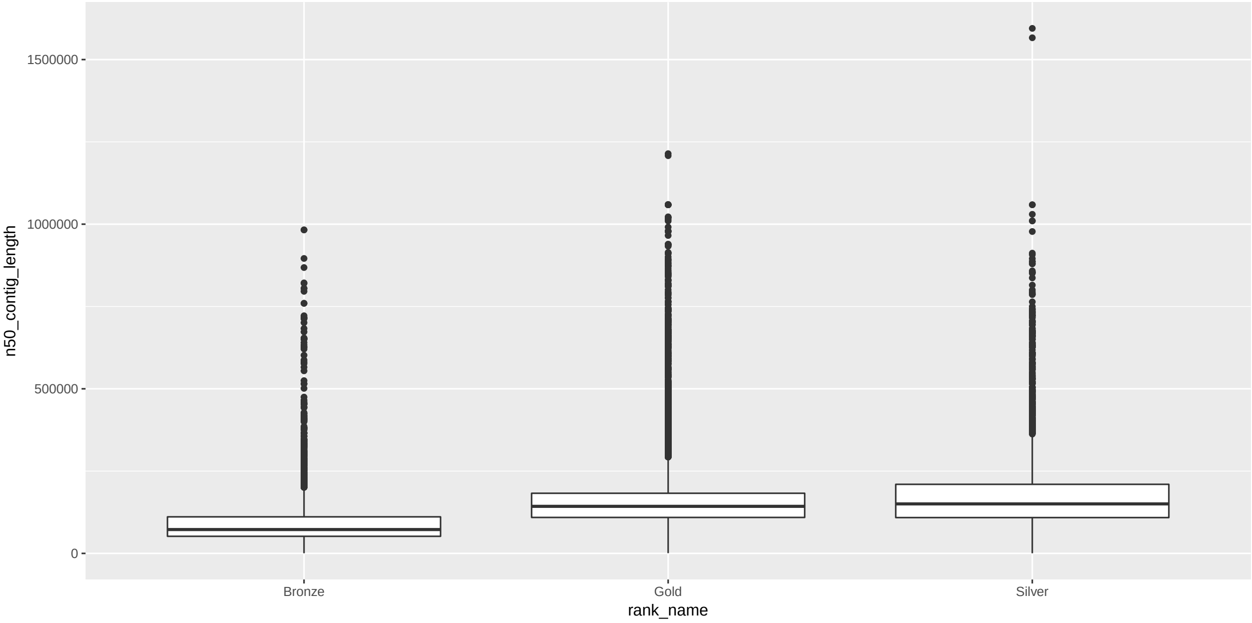
Total Contigs (smaller is better)

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



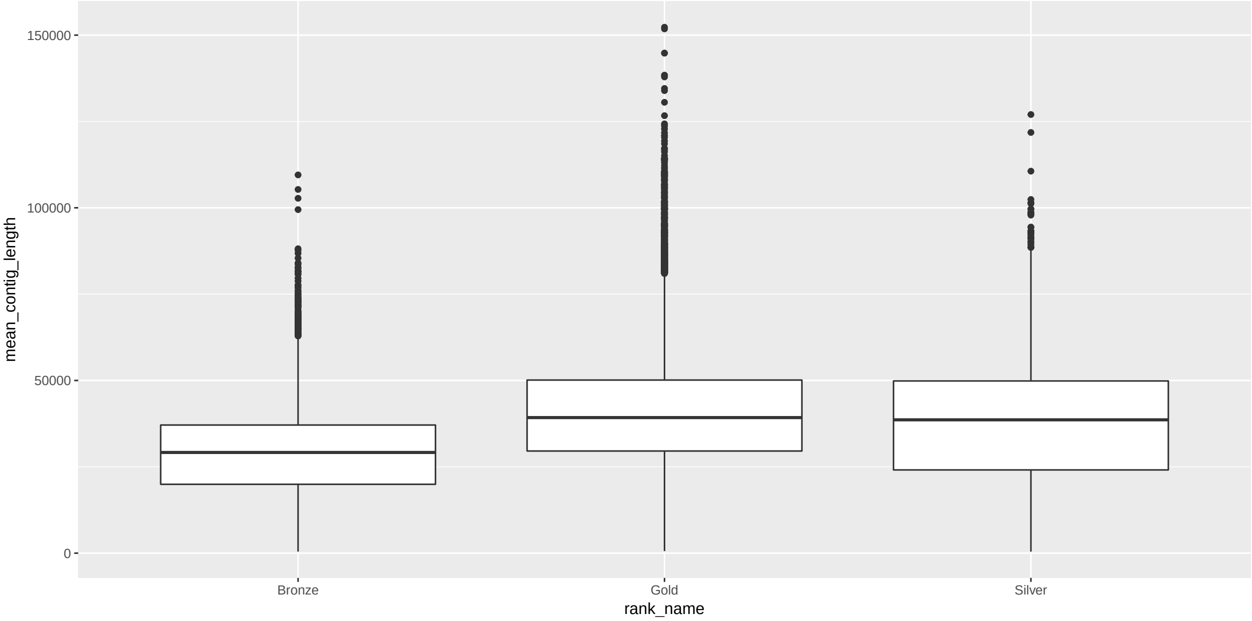
N50

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



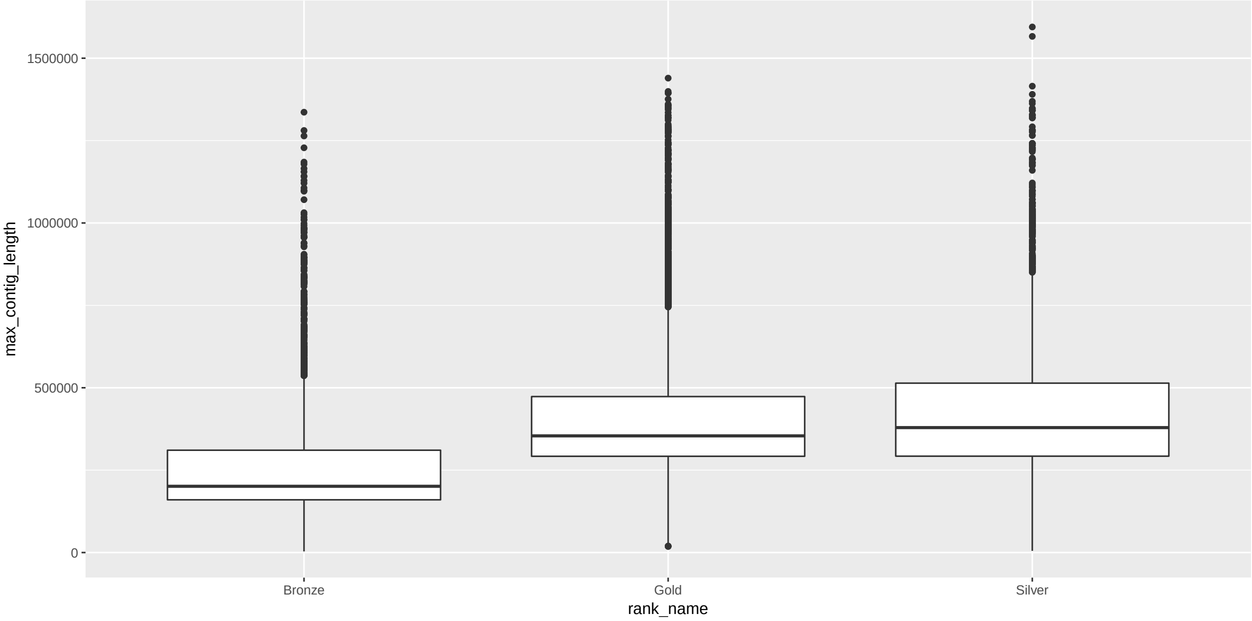
Mean Contig Length

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



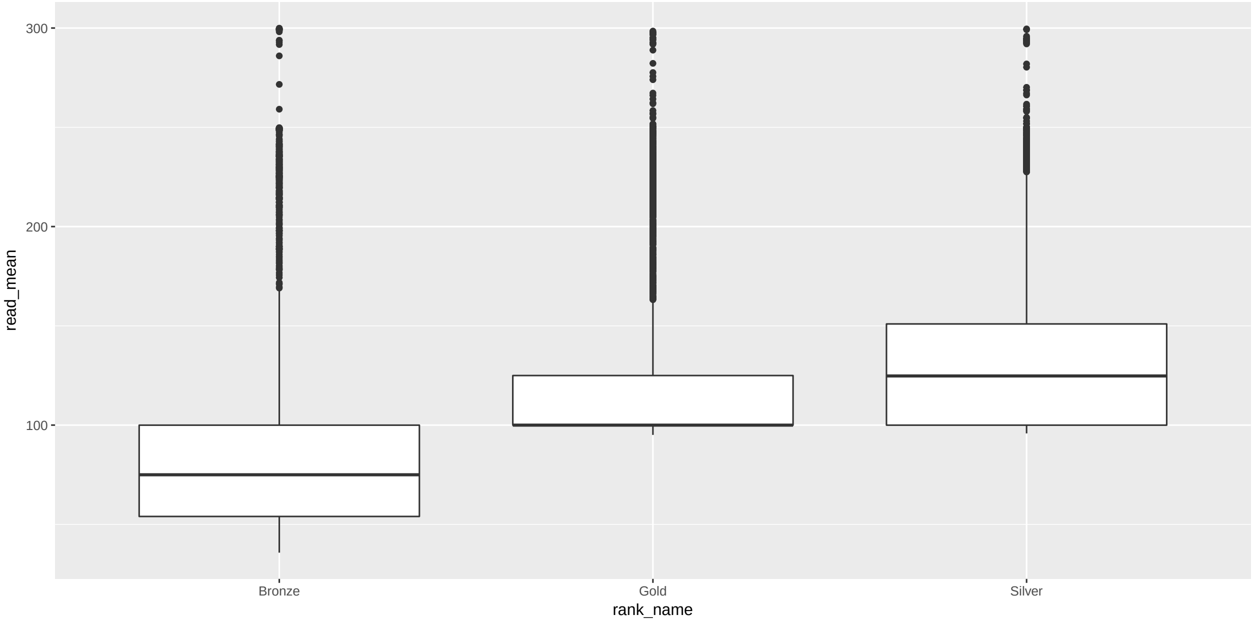
Max Contig Length

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



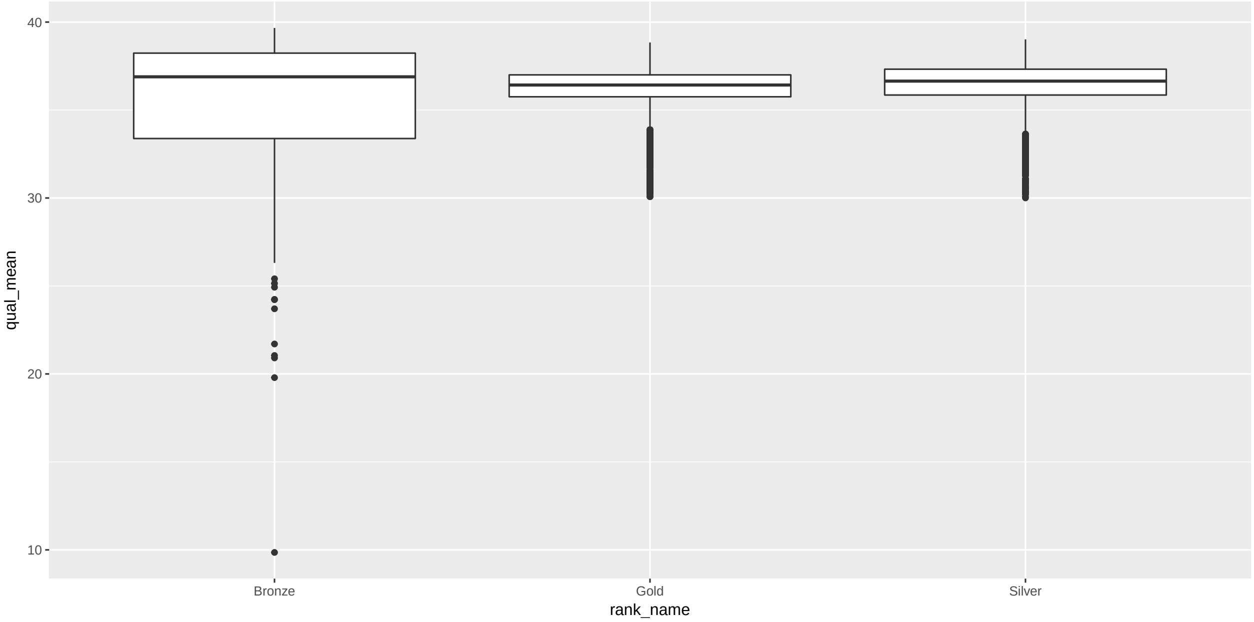
Mean Read Length

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



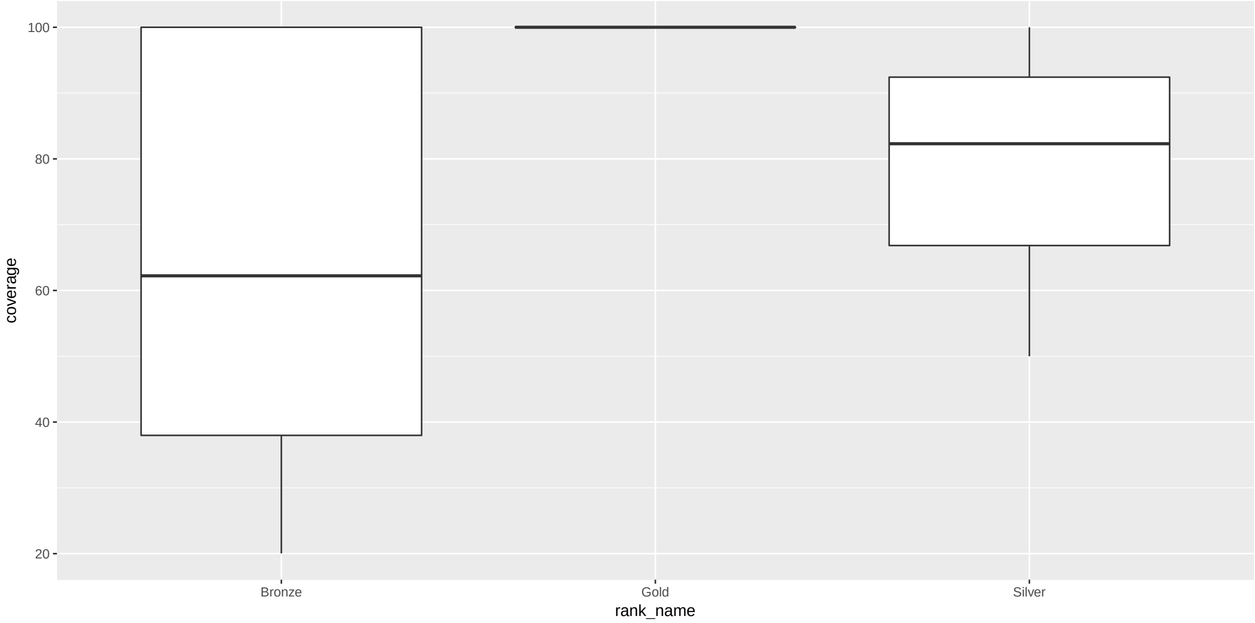
Mean Per-Read Quality Score

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



Coverage

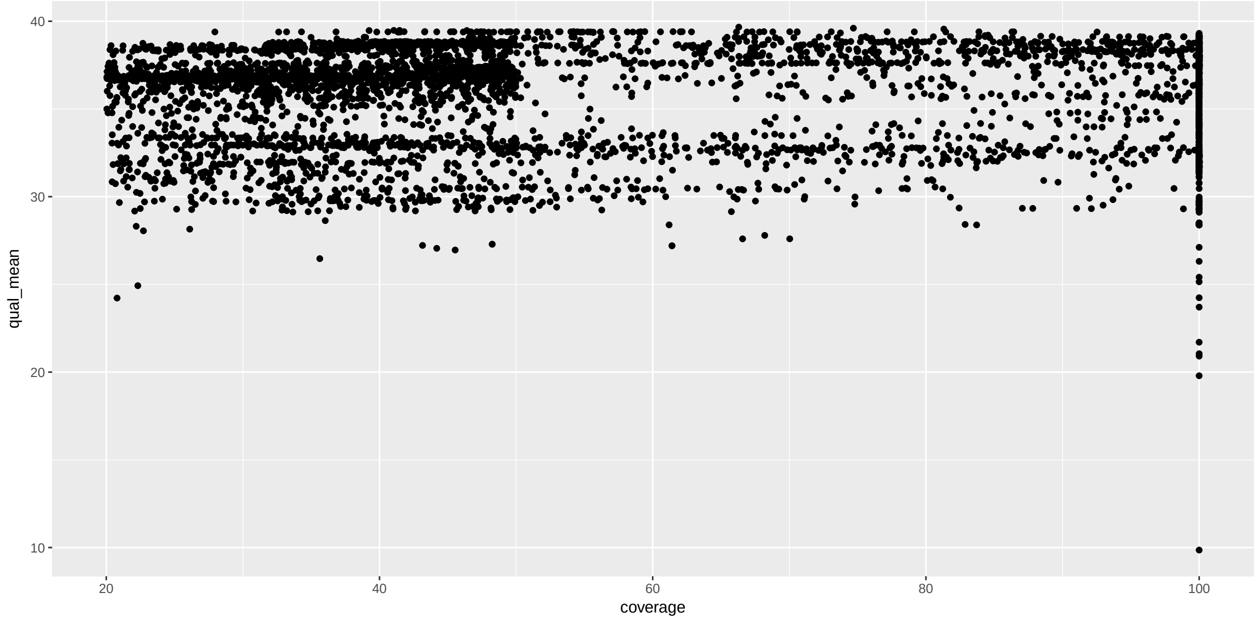
```
p <- ggplot(metrics, aes(x = rank_name, y = coverage)) +
    geom_boxplot()
p
```



Bronze Data

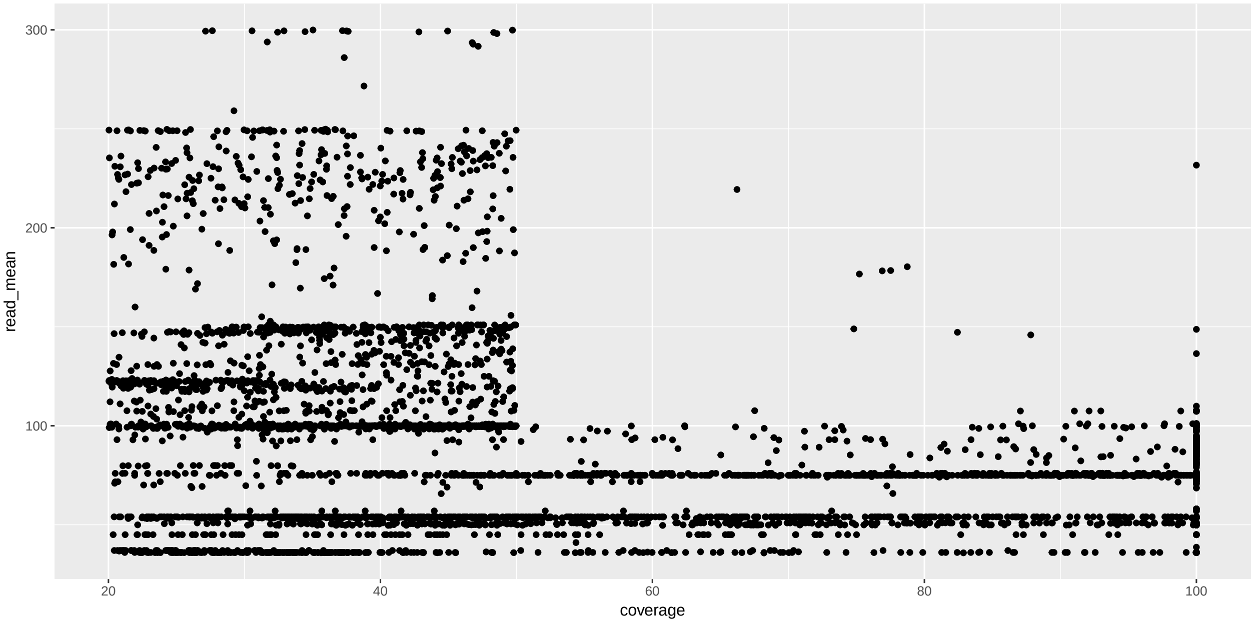
Coverage By Quality

```
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
```


read_mean
coverage