Data Analysis Project

Naoko Ishibashi

10/19/2024

Packages

```
#Packages
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(reshape2)
```

Data cleaning and checking

```
#import TSV file into data frame
HPAP001 <- read_tsv("../data/HPAP001.pooled.tsv")
```

```
## Rows: 26463 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (7): subject, v_gene, j_gene, cdr3_nt, cdr3_aa, germline, top_copy_seq
## dbl (6): clone_id, cdr3_num_nts, uniques, instances, copies, avg_v_identity
## lgl (4): functional, insertions, deletions, parent_id
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
HPAP003 <- read_tsv("../data/HPAP003.pooled.tsv")
```

```
## Rows: 17505 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (7): subject, v_gene, j_gene, cdr3_nt, cdr3_aa, germline, top_copy_seq
## dbl (6): clone_id, cdr3_num_nts, uniques, instances, copies, avg_v_identity
## lgl (4): functional, insertions, deletions, parent_id
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
HPAP004 <- read_tsv("../data/HPAP004.pooled.tsv")
```

```
## Rows: 16222 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (7): subject, v_gene, j_gene, cdr3_nt, cdr3_aa, germline, top_copy_seq
## dbl (6): clone_id, cdr3_num_nts, uniques, instances, copies, avg_v_identity
## lgl (4): functional, insertions, deletions, parent_id
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
HPAP015 <- read_tsv("../data/HPAP015.pooled.tsv")
```

```
## Rows: 47581 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (7): subject, v_gene, j_gene, cdr3_nt, cdr3_aa, germline, top_copy_seq
## dbl (6): clone_id, cdr3_num_nts, uniques, instances, copies, avg_v_identity
## lgl (4): functional, insertions, deletions, parent_id
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
HPAP030 <- read_tsv("../data/HPAP030.pooled.tsv")
```

```
## Rows: 45213 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (7): subject, v_gene, j_gene, cdr3_nt, cdr3_aa, germline, top_copy_seq
## dbl (6): clone_id, cdr3_num_nts, uniques, instances, copies, avg_v_identity
## lgl (4): functional, insertions, deletions, parent_id
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
HPAP031 <- read_tsv("../data/HPAP031.pooled.tsv")
```

```
## Rows: 36831 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (7): subject, v_gene, j_gene, cdr3_nt, cdr3_aa, germline, top_copy_seq
## dbl (6): clone_id, cdr3_num_nts, uniques, instances, copies, avg_v_identity
## lgl (4): functional, insertions, deletions, parent_id
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# Merge the datasets because HPAP001, HPAP003, HPAP004, HPAP015, HPAP030,
# and HPAP031 have the same columns
all_hpap <- rbind(HPAP001, HPAP003, HPAP004, HPAP015, HPAP030,HPAP031)

colSums(is.na(all_hpap))
```

```
##       clone_id        subject         v_gene         j_gene     functional
##              0              0              0              0              0
##     insertions      deletions        cdr3_nt   cdr3_num_nts        cdr3_aa
##         189815         189815              0              0              0
##        uniques      instances         copies       germline      parent_id
##              0              0              0              0         189815
## avg_v_identity   top_copy_seq
##              0              0
```

```r
# clean the data
# Delete NA values from 'insertions', 'deletions', and 'parent_id'
hpap_cleaned <- subset(all_hpap, select = -c(insertions, deletions, parent_id))
# check
colSums(is.na(hpap_cleaned))
```

```
##       clone_id        subject         v_gene         j_gene     functional
##              0              0              0              0              0
##        cdr3_nt   cdr3_num_nts        cdr3_aa        uniques      instances
##              0              0              0              0              0
##         copies       germline avg_v_identity   top_copy_seq
##              0              0              0              0
```

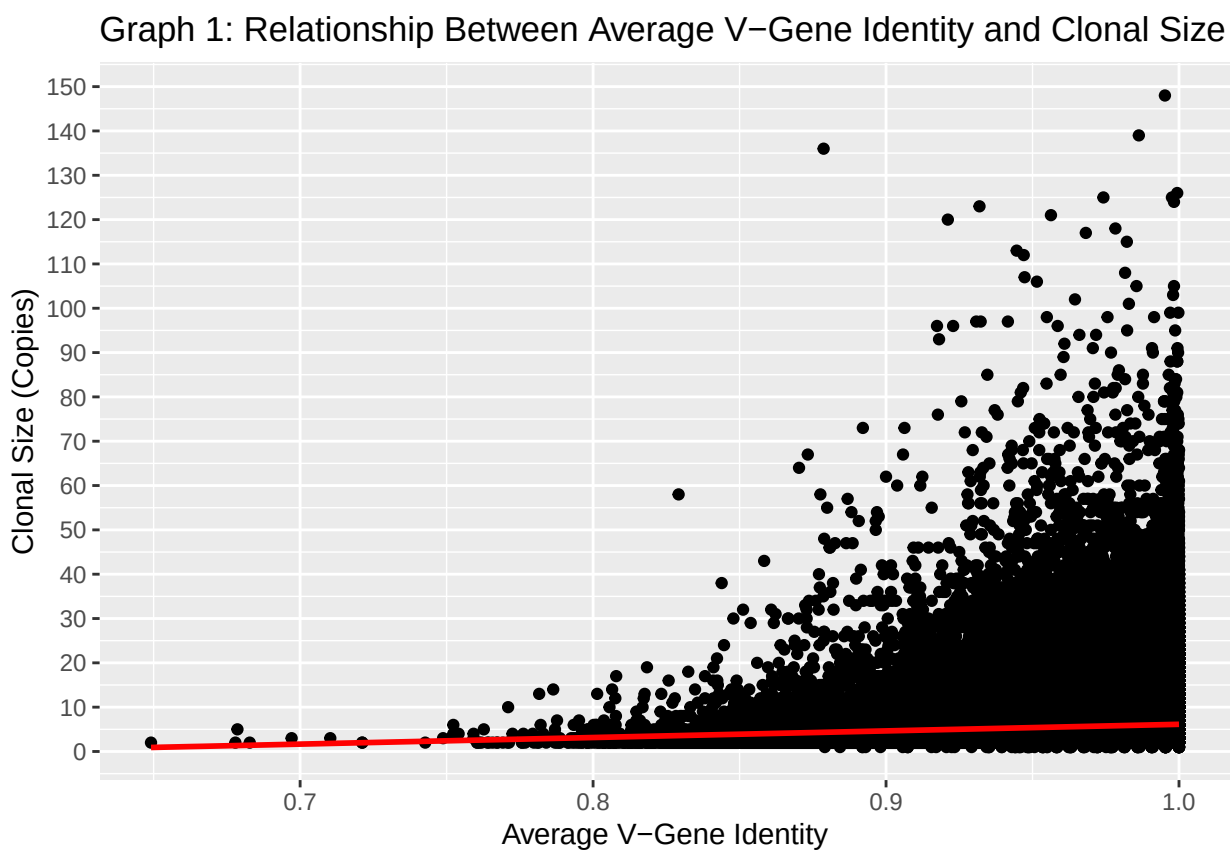
1) Is there a correlation between clone size and mutation?

When a B-cell activates in the presence of antigenic stimulation, it rapidly multiplies and mutates. For each clone we have two fields avg_v_identity and copies. The former gives the fraction of nucleotide bases in the clone that match the germline (e.g. 1 - mutation) and the latter gives the number of reads associated with the clone, giving a rough measure of size. Given this, come up with a figure that one could use to see if there is a correlation between these two variables. Next, statistically test the hypothesis that the two are related and comment on your choice of statistical test and its robustness.

1. Given this, come up with a figure that one could use to see if there is a correlation between these two variables.
2. Next, statistically test the hypothesis that the two are related and comment on your choice of statistical test and its robustness.

```r
# Create a data visualization to examine the correlation between these two variables.
# Assuming df is your data frame and y_var has the potential outliers
cleaned_data <- hpap_cleaned %>% filter (copies < 150) # Remove outliers


# Create the scatter plot: A scatter plot is able to display the relationship
# between two variables effectively.
# Graph 1: Scatter plot of Average V-Gene Identity vs. Clonal Size
ggplot(data = cleaned_data, aes(x = avg_v_identity, y = copies)) +
  geom_point() +
  geom_smooth(method = "lm", color = "red", se = FALSE) +  # Add regression line
  scale_y_continuous(breaks = seq(0, 150, by = 10)) +  # Add breaks
  labs(title = "Graph 1: Relationship Between Average V-Gene Identity and Clonal Size",
       x = "Average V-Gene Identity",
       y = "Clonal Size (Copies)")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Analysis (Graph)

This graph shows a positive trend. The points start to gather more around 0.75 on the x-axis, with most points between (0.85, 10) and (1, 40). The slope is set to 2 to make it easier to see, but the real trend is a gentle upward slope."

```r
# Statistically test the hypothesis that clone size and mutations are related.
# The cor() function is used to calculate the correlation coefficient between two variables.

# Pearson
# Calculate the Pearson Correlation: measures linear relationships.
correlation_pearson <- cor(hpap_cleaned$avg_v_identity, hpap_cleaned$copies, method = "pearson")
print(correlation_pearson) # 0.0649304
```

```
## [1] 0.0649304
```

```r
# Spearman
# Calculate the Spearman Correlation: measures rank-based relationships.
correlation_spearman <- cor(hpap_cleaned$avg_v_identity, hpap_cleaned$copies, method = "spearman")
print(correlation_spearman) # -0.02137582
```

```
## [1] -0.02137582
```

```r
# Kendall
# Calculate the Kendall Correlation: measures ordinal relationships.
correlation_kendall <- cor(hpap_cleaned$avg_v_identity, hpap_cleaned$copies, method = "kendall")
print(correlation_kendall) # -0.02137582
```

```
## [1] -0.01497234
```

Analysis (Pearson,Spearman or Kendall)

I believe that the Pearson Correlation is the appropriate choice for this analysis. The scatter plot clearly shows a positive trend, and the Pearson Correlation reflects that positivity. For these reasons, I consider the Pearson Correlation the most reliable option.

```r
# Perform correlation test
correlation_test <- cor.test(hpap_cleaned$copies, hpap_cleaned$avg_v_identity, method = "pearson")
print(correlation_test)
```

```
##
##  Pearson's product-moment correlation
##
## data:  hpap_cleaned$copies and hpap_cleaned$avg_v_identity
## t = 28.348, df = 189813, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.06044939 0.06940878
## sample estimates:
##       cor
## 0.0649304
```

Analysis (correlation test)

t(Test Statistic) = 28.348, The strength of the relationship between the two variables.

df (Degrees of Freedom) = 189813, The size of the sample used in the analysis.

p-value < 2.2e-16 The p-value is less than 2.2e-16; this is significantly lower than the conventional alpha level of 0.05. This indicates that we can reject the null hypothesis, providing strong evidence of a correlation between the two variables.

Alternative Hypothesis: There is a significant relationship between copies and avg_v_identity.

95 percent confidence interval: 0.06044939 to 0.06940878. This number indicates a positive connection, meaning that as one thing increases, the other also tends to increase.

sample estimates: cor 0.0649304 This value indicates a positive connection.

```r
# linear
# Find the strength of the association (slope) and examine the significance of this relationship.
linear <- summary(lm(avg_v_identity ~ copies, data = hpap_cleaned))

# linear
print(linear)
```

```
##
## Call:
## lm(formula = avg_v_identity ~ copies, data = hpap_cleaned)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32691 -0.01682  0.01657  0.02370  0.02428
##
## Coefficients:
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept) 9.754e-01  9.434e-05 10339.23   <2e-16 ***
## copies      2.873e-04  1.014e-05    28.35   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03211 on 189813 degrees of freedom
## Multiple R-squared:  0.004216,   Adjusted R-squared:  0.004211
## F-statistic: 803.6 on 1 and 189813 DF,  p-value: < 2.2e-16
```

Analysis (linear)

A p-value of < 2.2e-16 indicates a highly significant relationship between copy number and avg_v_identity. This extremely low p-value strongly suggests that we can reject the null hypothesis, meaning the likelihood of observing this relationship by chance is almost zero. The results show a statistically significant positive correlation between copy number and avg_v_identity. In other words, as clone size (measured by copy number) increases, the average V-gene identity (avg_v_identity) also tends to increase.

2) Is V-gene usage uniform within and consistent between donors?

During development, each B-cell undergoes V(D)J recombination that pseudo-randomly joins one V, one D, and one J gene to create part of the heavy-chain which goes on to encode a portion of the cell's antibodies. For this question we'll focus on the V-gene which is annotated by the v_gene column. Does it appear that within each donor V-genes are evenly distributed amongst clones? How would you visualize this? What happens when you look at V-gene usage weighted by copies? Across donors, does it appear that there is a pattern of V-gene usage (i.e. do most donors use the same V-genes) or is each donor different? How would you visualize this?

1, Does it appear that within each donor V-genes are evenly distributed amongst clones? How would you visualize this? What happens when you look at V-gene usage weighted by copies?

2, Across donors, does it appear that there is a pattern of V-gene usage (i.e. do most donors use the same V-genes) or is each donor different? How would you visualize this?

```
# 1.
# Check the data.
table(hpap_cleaned$subject)
```

```
##
## HPAP001 HPAP003 HPAP004 HPAP015 HPAP030 HPAP031
##   26463   17505   16222   47581   45213   36831
```

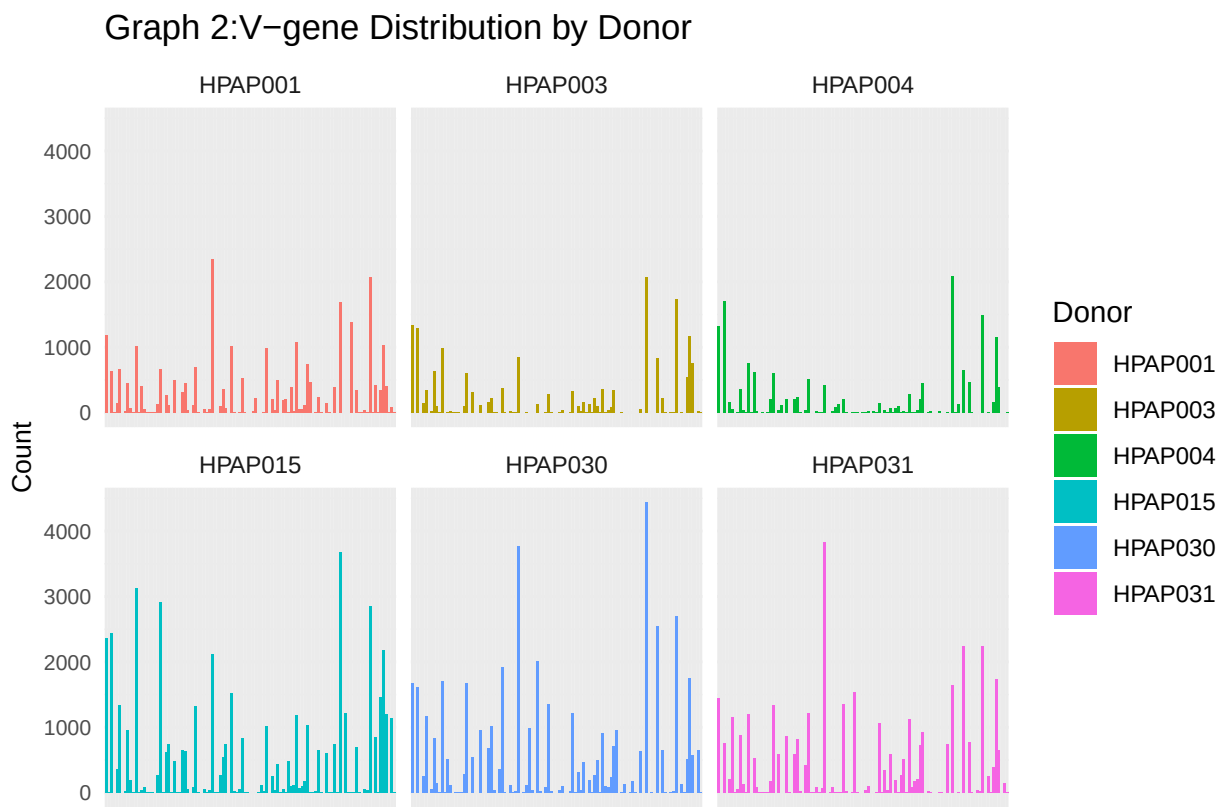
```
str(hpap_cleaned$v_gene)
```

```
##  chr [1:189815] "IGHV1-18" "IGHV1-18" "IGHV1-18" "IGHV1-18" "IGHV1-18" ...
```

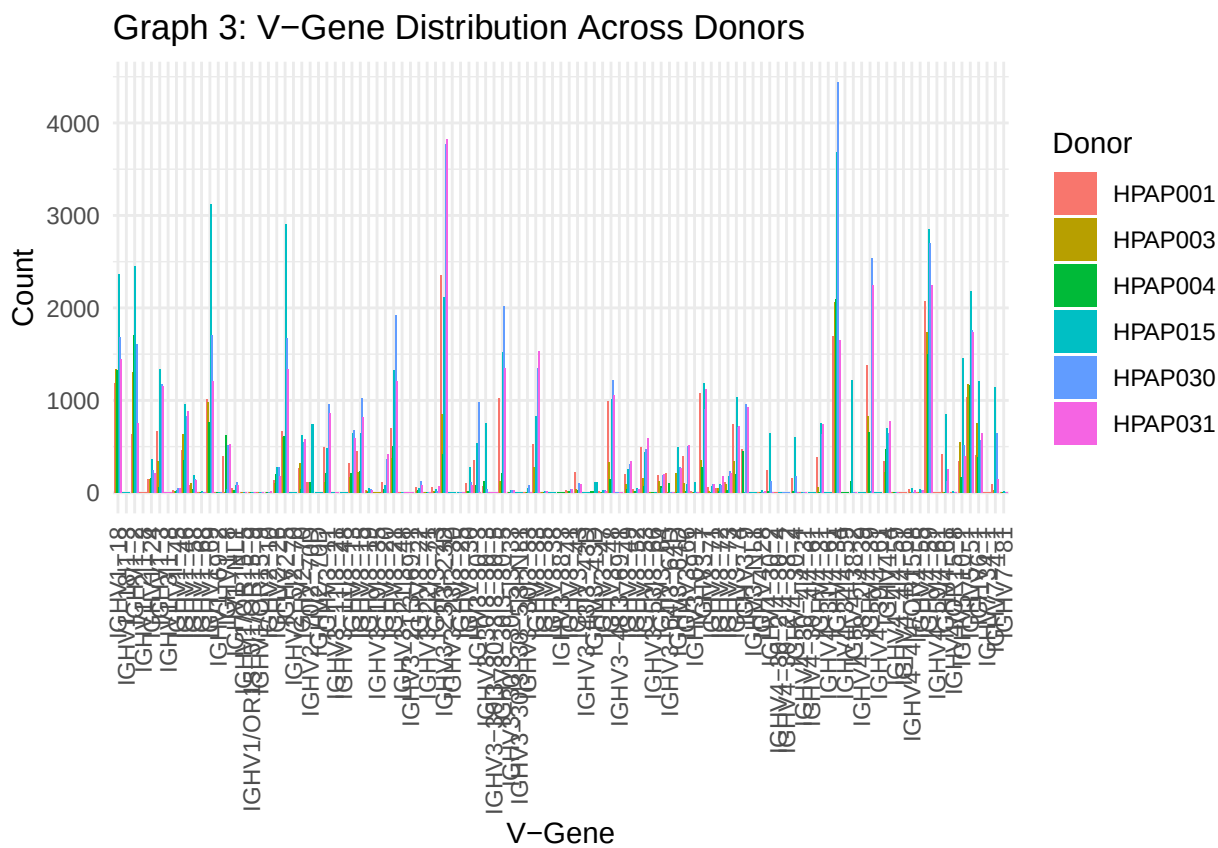
```
# Group data by donor (subject) and count the occurrences of each V-gene within each donor:
v_gene_summary <- hpap_cleaned %>%
  group_by(subject, v_gene) %>%    # Group by donors and V-gene
  summarise(
    v_gene_count = n(),            # Count how many times each V-gene appears for that donor
    weighted_copies = sum(copies)  # Add up the values to calculate the weighted copies
  )
```

```
## `summarise()` has grouped output by 'subject'. You can override using the
## `.groups` argument.
```

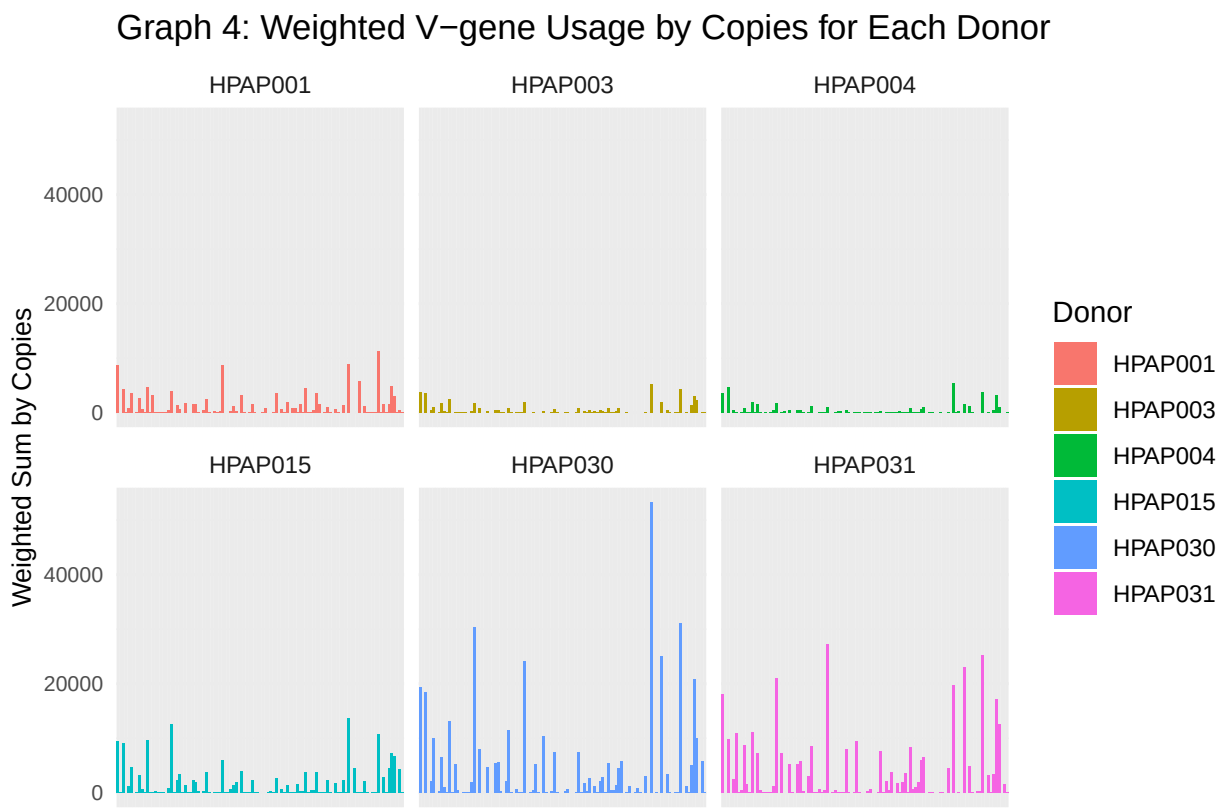
```
# Graph 2: Bar plots showing the distribution of V-genes for each donor.
ggplot(v_gene_summary, aes(x = v_gene, y = v_gene_count, fill = as.factor(subject))) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Graph 2:V-gene Distribution by Donor",
       x = "",  # Remove the x-axis label
       y = "Count",
       fill = "Donor") +
  theme_minimal() +
  theme(axis.text.x = element_blank(),  # Remove x-axis text
        axis.ticks.x = element_blank(),  # Remove x-axis ticks
        axis.text.y = element_text(size = 8),  # Keep y-axis text size
        axis.title.y = element_text(size = 10)) +  # Optional: Adjust y-axis title size
  facet_wrap(~ subject)
```



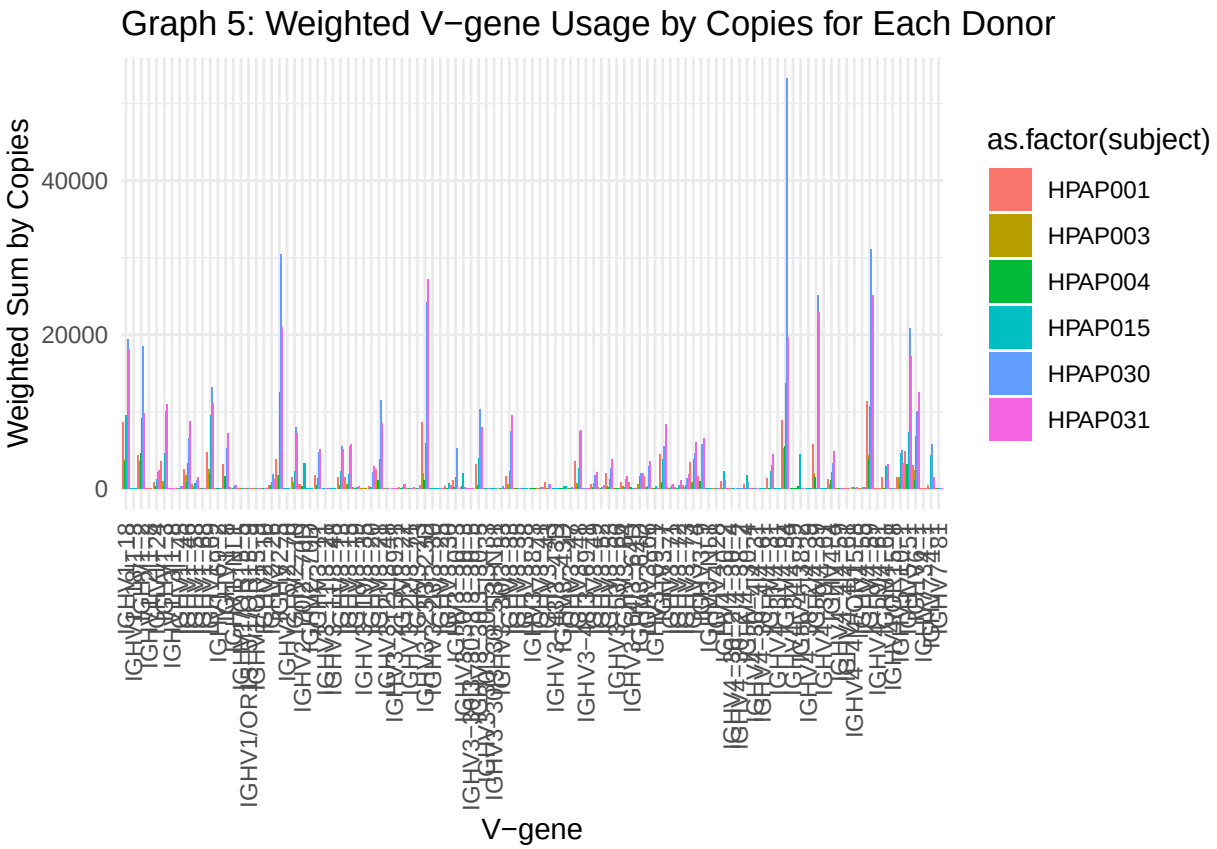

```r
# Graph 3: Bar plots showing the distribution of V-genes across donors.
ggplot(v_gene_summary, aes(x = v_gene, y = v_gene_count, fill = subject)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Graph 3: V-Gene Distribution Across Donors",
       x = "V-Gene",
       y = "Count",
       fill = "Donor") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```




```
# Weighted V-gene Distribution
# Graph 4: Plot of V-gene distribution weighted by copies for each donor.
ggplot(v_gene_summary, aes(x = v_gene, y = weighted_copies, fill = as.factor(subject))) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Graph 4: Weighted V-gene Usage by Copies for Each Donor",
       x = "",  # Remove the x-axis label
       y = "Weighted Sum by Copies",
       fill = "Donor") +
  theme_minimal() +
  theme(axis.text.x = element_blank(),  # Remove x-axis text
        axis.ticks.x = element_blank(),  # Remove x-axis ticks
        axis.text.y = element_text(size = 8),  # Keep y-axis text size
        axis.title.y = element_text(size = 10)) +  # Optional: Adjust y-axis title size
  facet_wrap(~ subject)
```




```r
# Weighted V-gene Distribution
# Graph 5: Plot of V-gene distribution weighted by copies for each donor.
ggplot(v_gene_summary, aes(x = v_gene, y = weighted_copies, fill = as.factor(subject))) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Graph 5: Weighted V-gene Usage by Copies for Each Donor",
       x = "V-gene",
       y = "Weighted Sum by Copies") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```



Analysis (Graph 2-5)

Without weighting by copies: In Graph 2: - HPAP003 and HPAP004 show similar patterns. - The V-gene count values for HPAP015, HPAP030, and HPAP031 are higher than those of HPAP001, HPAP003, and HPAP004.

In Graph 3: - The V-gene count values vary by donor but overall follow a similar flow. - HPAP015 stands out with significantly higher V-gene count values than the others, particularly in: - IGHV1-18. - IGHV1-2. - IGHV1-69. - IGHV2-5. - IGHV4-34. - HPAP030 and HPAP015 have significantly higher V-gene count values for IGHV4-34. - HPAP030 and HPAP031 have significantly higher V-gene count values for: - IGHV2-5. - IGHV3-23 | 3-23D.

With weighting by copies: In Graph 4: - After weighting by copies, the overall V-gene count values are lower than in the unweighted graph. - HPAP003 and HPAP004 maintain a similar pattern as before. - In contrast, HPAP030 and HPAP031 now show higher V-gene count values than the others, making the differences more pronounced. - HPAP001 and HPAP015 display similar wave patterns.

In Graph 5: - V-gene count values vary across donors, but the overall patterns remain consistent. HPAP030 and HPAP031 have significantly higher copy values than the others, particularly in: - IGHV2-5. - IGHV3-23 | 3-23D. - IGHV4-34. - IGHV4-39. - IGHV4-59.

Conclusion: - When comparing the weighted and unweighted data, the V-gene count values are generally lower after weighting by copies. - Notably, HPAP030 and HPAP031 show dramatically higher values for IGHV2-5 and IGHV3-23 | 3-23D compared to the other donors. - Additionally, IGHV4-34 in HPAP030 remains consistently higher overall.

3) Does disease affect the clonal repertoire?

In this final task, we introduce one additional piece of information. HPAP001, HPAP003, and HPAP004 are all serologically healthy individuals whereas HPAP015, HPAP030, and HPAP031 have all been clinically diagnosed with type 1 diabetes. Given this, see if you can find a difference between controls and those with diabetes. You may use any method and any fields you'd like, but please make at least one or two figures to present your findings. Simply looking at individual columns will likely be insufficient, so consider how combinations of fields can discriminate between disease and health.

```
# Create a new column based on the subject column that distinguishes between healthy individuals and th
hpap_cleaned$health_status[hpap_cleaned$subject == "HPAP001"] <- "Healthy"
```

```
## Warning: Unknown or uninitialised column: `health_status`.
```

```
hpap_cleaned$health_status[hpap_cleaned$subject == "HPAP003"] <- "Healthy"
hpap_cleaned$health_status[hpap_cleaned$subject == "HPAP004"] <- "Healthy"

hpap_cleaned$health_status[hpap_cleaned$subject == "HPAP015"] <- "Diabetes"
hpap_cleaned$health_status[hpap_cleaned$subject == "HPAP030"] <- "Diabetes"
hpap_cleaned$health_status[hpap_cleaned$subject == "HPAP031"] <- "Diabetes"

# Create a new column based on the subject column that distinguishes between healthy individuals and in
# Check the column names
colnames(hpap_cleaned) # check
```

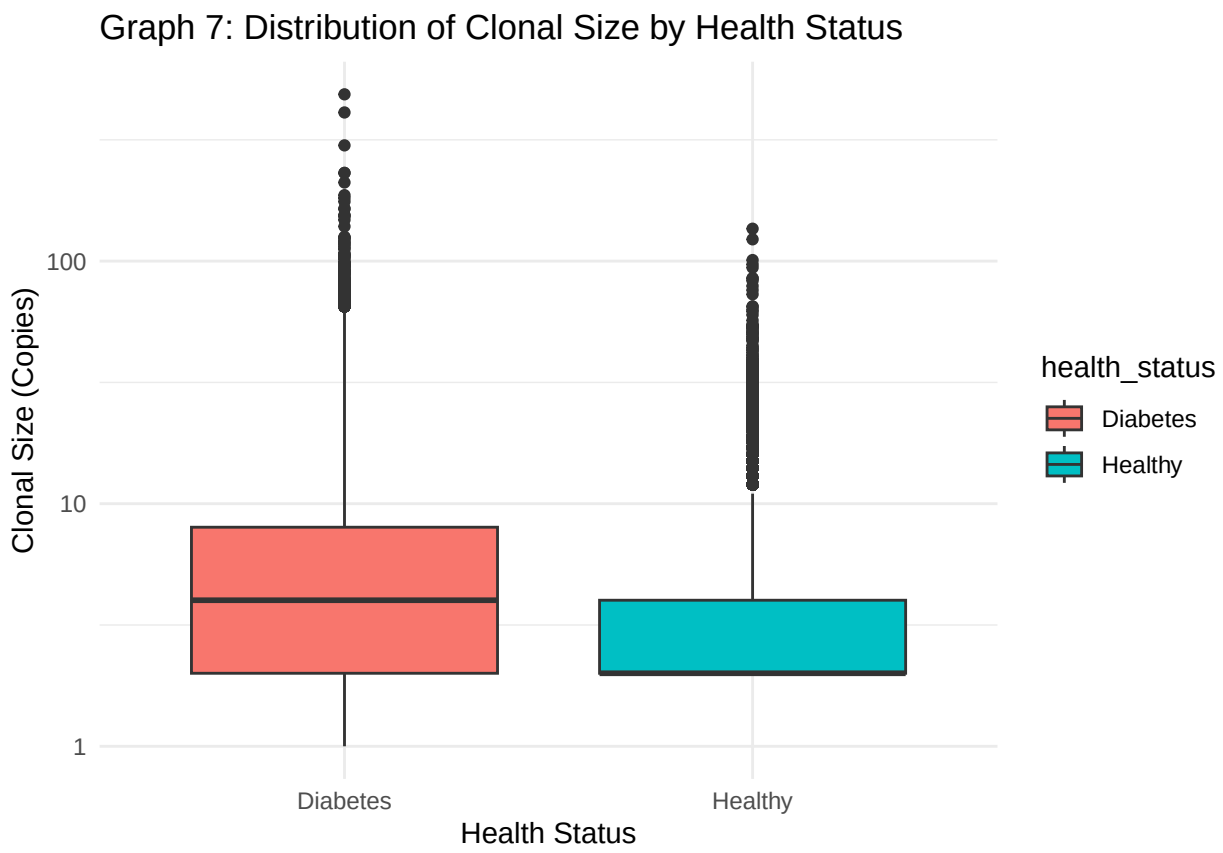
```
##  [1] "clone_id"       "subject"        "v_gene"         "j_gene"
##  [5] "functional"     "cdr3_nt"        "cdr3_num_nts"   "cdr3_aa"
##  [9] "uniques"        "instances"      "copies"         "germline"
## [13] "avg_v_identity" "top_copy_seq"   "health_status"
```

```
# Graph 6: Visualizing V-Gene Usage Across Health Status
ggplot(data = hpap_cleaned, aes(x = v_gene, fill = health_status)) +
  geom_bar(position = "dodge") +
  labs(title = "Graph 6: V-Gene Usage Across Health Status",   # Adding the title here
       x = "V-Gene",
       y = "Count",
       fill = "Health Status") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```

Graph 6: Visualizing V-Gene Usage Across Health Status Analysis.

Overall, the V-gene counts of clones in individuals with diabetes are higher than in healthy individuals.

```
# Graph 7: Box Plot
# The distribution was difficult to understand because of the large outlier in "copies."
# So, I used a log transformation to make it easier to view.
ggplot(data = hpap_cleaned, aes(x = health_status, y = copies, fill = health_status)) +
  geom_boxplot() +
  scale_y_log10() +  # Adjusted the Y-Axis Scale to make it easier to view the distribution.
  labs(title = "Graph 7: Distribution of Clonal Size by Health Status",
       x = "Health Status",
       y = "Clonal Size (Copies)") +
  theme_minimal()
```



Graph 7: Box Plot Analysis

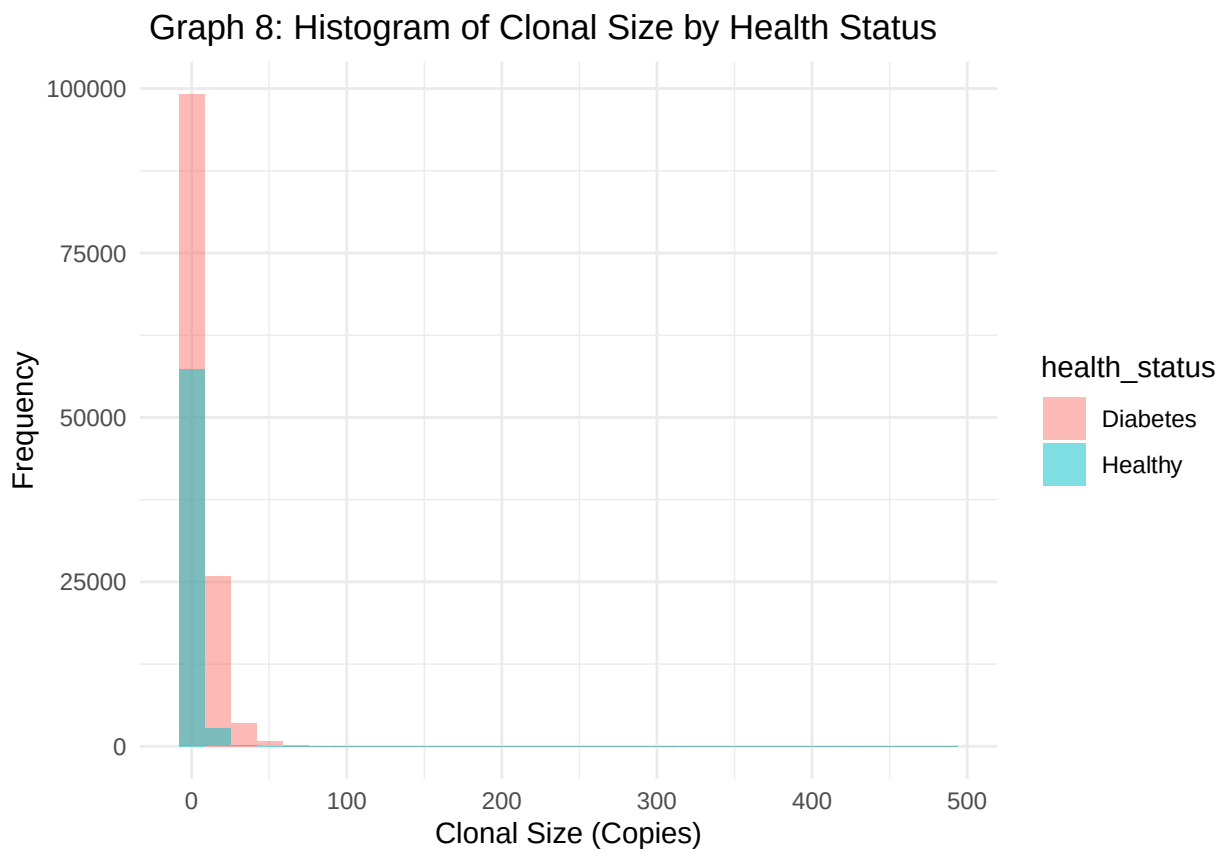
I used a box plot to compare the clonal size of individuals with diabetes and healthy individuals.

Diabetes: - The position of the box is lower. - The interquartile range (IQR) spans from around 3 (25%) to 8 (75%). - The median is approximately 6. - The data is less clustered compared to healthy samples.

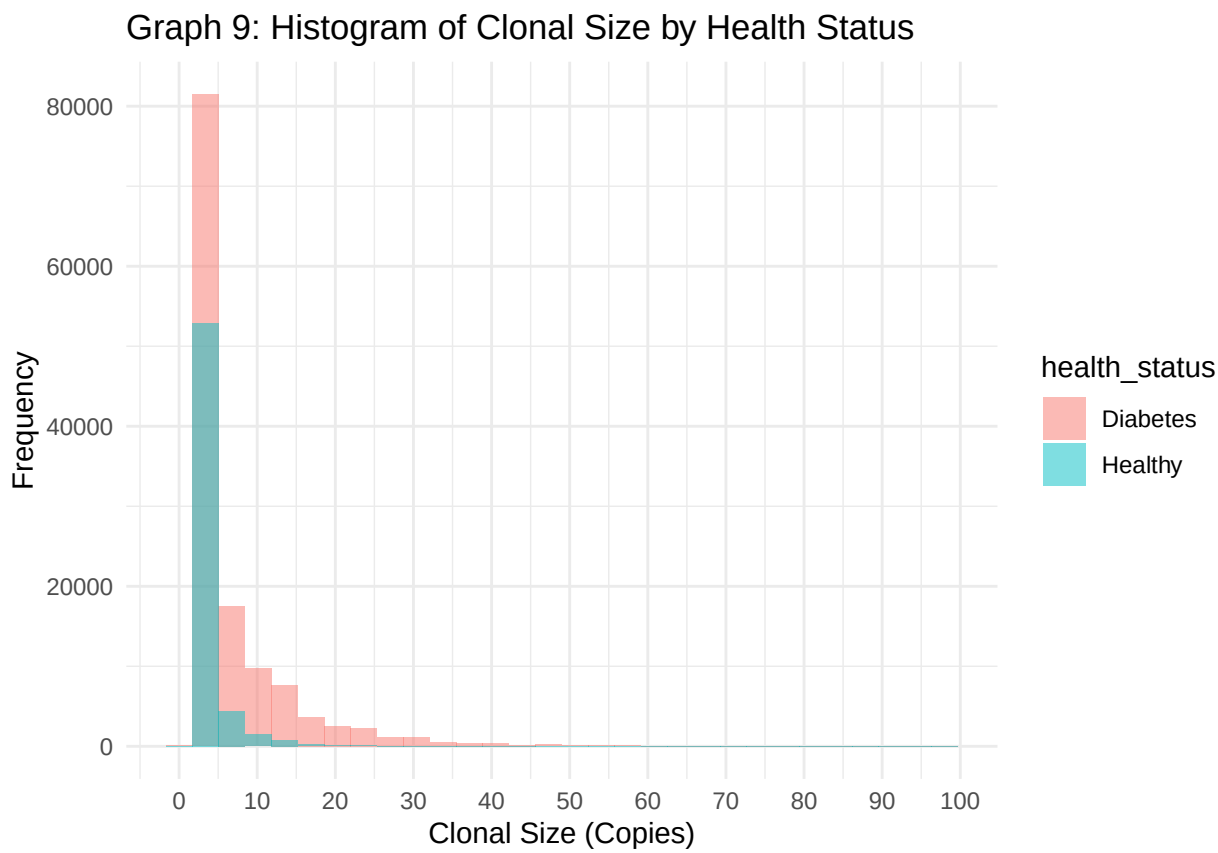
Healthy: - The position of the box is higher. - The IQR spans from about 3 (25%) to 6 (75%). - The median is around 3. - The data is more clustered compared to the diabetes group.

In summary: The clonal size in individuals with diabetes tends to be larger than in healthy individuals. The median clonal size for healthy individuals is smaller at around 3, compared to 6 for individuals with diabetes.

```
# Graph 8: Histogram
ggplot(data = hpap_cleaned, aes(x = copies, fill = health_status)) +
  geom_histogram(position = "identity", alpha = 0.5, bins = 30) +
  labs(title = " Graph 8: Histogram of Clonal Size by Health Status",
       x = "Clonal Size (Copies)",
       y = "Frequency") +
  theme_minimal()
```




```r
# Removing outliers makes the graph easier to view.
cleaned_data2 <- hpap_cleaned %>% filter(copies < 100) # Remove outliers

# Graph 9: Histogram (A Closer Examination)
ggplot(data = cleaned_data2, aes(x = copies, fill = health_status)) +
  geom_histogram(position = "identity", alpha = 0.5, bins = 30) +
  labs(title = "Graph 9: Histogram of Clonal Size by Health Status",
       x = "Clonal Size (Copies)",
       y = "Frequency") +
  scale_x_continuous(breaks = seq(0, 100, by = 10)) +  # Set breaks on the x-axis
  theme_minimal()
```




```r
# Count occurrences of each health status
health_count <- table(hpap_cleaned$health_status)
# Table 10: Display the counts
print(health_count)
```

```
##
## Diabetes  Healthy
##   129625    60190
```

```r
# Summary statistics for clonal sizes by health status
dis_table <- by(hpap_cleaned$copies, hpap_cleaned$health_status, summary)
# Table 11
print(dis_table)
```

```
## hpap_cleaned$health_status: Diabetes
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   4.000   6.902   8.000 487.000
## ------------------------------------------------------------
## hpap_cleaned$health_status: Healthy
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   2.000   2.000   3.458   4.000 136.000
```

Analysis(Graph 8, 9,)

- Graph 8 displays the distribution of clonal sizes for both healthy individuals and those with type 1 diabetes. It is evident from the graph that there are significantly more clones in individuals with type 1 diabetes than in healthy individuals.
- Graph 9 provides a closer examination of the clonal distribution, showing that type 1 diabetic clones are both larger in number and size compared to healthy clones.
- Table 10: summarizes the count of clones for each group, with 129,625 clones for individuals with diabetes and 60,190 clones for healthy individuals.
- Table 11 presents the summary statistics for clonal sizes based on health status. In the diabetic group, the minimum clonal size is 1, with a mean size of 6.902 and a maximum size of 487. In contrast, the healthy group has a minimum clonal size of 2, a mean size of 3.458, and a maximum size of 136.

In conclusion, the clonal number in type 1 diabetes is roughly double that of healthy individuals. Additionally, the data suggests that individuals with clonal sizes smaller than 2 or larger than 136 are more likely to have type 1 diabetes.

4) Is there a correlation between clone size and mutations in healthy individuals and those with type 1 diabetes? (Free investigation)

```r
# Correlation test
# Overall Correlation Test
correlation_test <- cor.test(hpap_cleaned$copies, hpap_cleaned$avg_v_identity, method = "pearson")
# Table 12: Overall Correlation Test
print(correlation_test)
```

```
##
##  Pearson's product-moment correlation
##
## data:  hpap_cleaned$copies and hpap_cleaned$avg_v_identity
## t = 28.348, df = 189813, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.06044939 0.06940878
## sample estimates:
##       cor
## 0.0649304
```

```r
# Healthy Group Correlation Test
correlation_test_healthy <- cor.test(hpap_cleaned$copies[hpap_cleaned$health_status == "Healthy"], hpap
# Table 13: Healthy Group Correlation Test
print(correlation_test_healthy)
```

```
##
##  Pearson's product-moment correlation
##
## data:  hpap_cleaned$copies[hpap_cleaned$health_status == "Healthy"] and hpap_cleaned$avg_v_identity[
## t = -30.386, df = 60188, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
```

```
## 95 percent confidence interval:
##  -0.1307792 -0.1150428
## sample estimates:
##        cor
## -0.1229187
```

```r
# Diabetes Group Correlation Test
correlation_test_diabetes <- cor.test(hpap_cleaned$copies[hpap_cleaned$health_status == "Diabetes"], hpa
# Table 14: Diabetes Group Correlation Test
print(correlation_test_diabetes)
```

```
##
##  Pearson's product-moment correlation
##
## data:  hpap_cleaned$copies[hpap_cleaned$health_status == "Diabetes"] and hpap_cleaned$avg_v_identity
## t = 36.829, df = 129623, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.09637261 0.10714752
## sample estimates:
##      cor
## 0.101763
```

```r
# Healthy Group Regression
healthy_model <- lm(avg_v_identity ~ copies, data = hpap_cleaned[hpap_cleaned$health_status == "Healthy
# Table 15: Healthy Group Regression
summary(healthy_model)
```

```
##
## Call:
## lm(formula = avg_v_identity ~ copies, data = hpap_cleaned[hpap_cleaned$health_status ==
##     "Healthy", ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.26626 -0.01846  0.02168  0.02219  0.12945
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.803e-01  1.966e-04 4987.43   <2e-16 ***
## copies      -1.256e-03  4.134e-05  -30.39   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0331 on 60188 degrees of freedom
## Multiple R-squared:  0.01511,    Adjusted R-squared:  0.01509
## F-statistic: 923.3 on 1 and 60188 DF,  p-value: < 2.2e-16
```

```r
# Diabetes Group Regression
diabetes_model <- lm(avg_v_identity ~ copies, data = hpap_cleaned[hpap_cleaned$health_status == "Diabet
# Table 16: Diabetes Group Regression
summary(diabetes_model)
```

```
##
## Call:
## lm(formula = avg_v_identity ~ copies, data = hpap_cleaned[hpap_cleaned$health_status ==
##     "Diabetes", ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32662 -0.01460  0.01495  0.02311  0.02467
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 9.749e-01  1.136e-04 8578.67   <2e-16 ***
## copies      3.880e-04  1.053e-05   36.83   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03145 on 129623 degrees of freedom
## Multiple R-squared:  0.01036,    Adjusted R-squared:  0.01035
## F-statistic:  1356 on 1 and 129623 DF,  p-value: < 2.2e-16
```

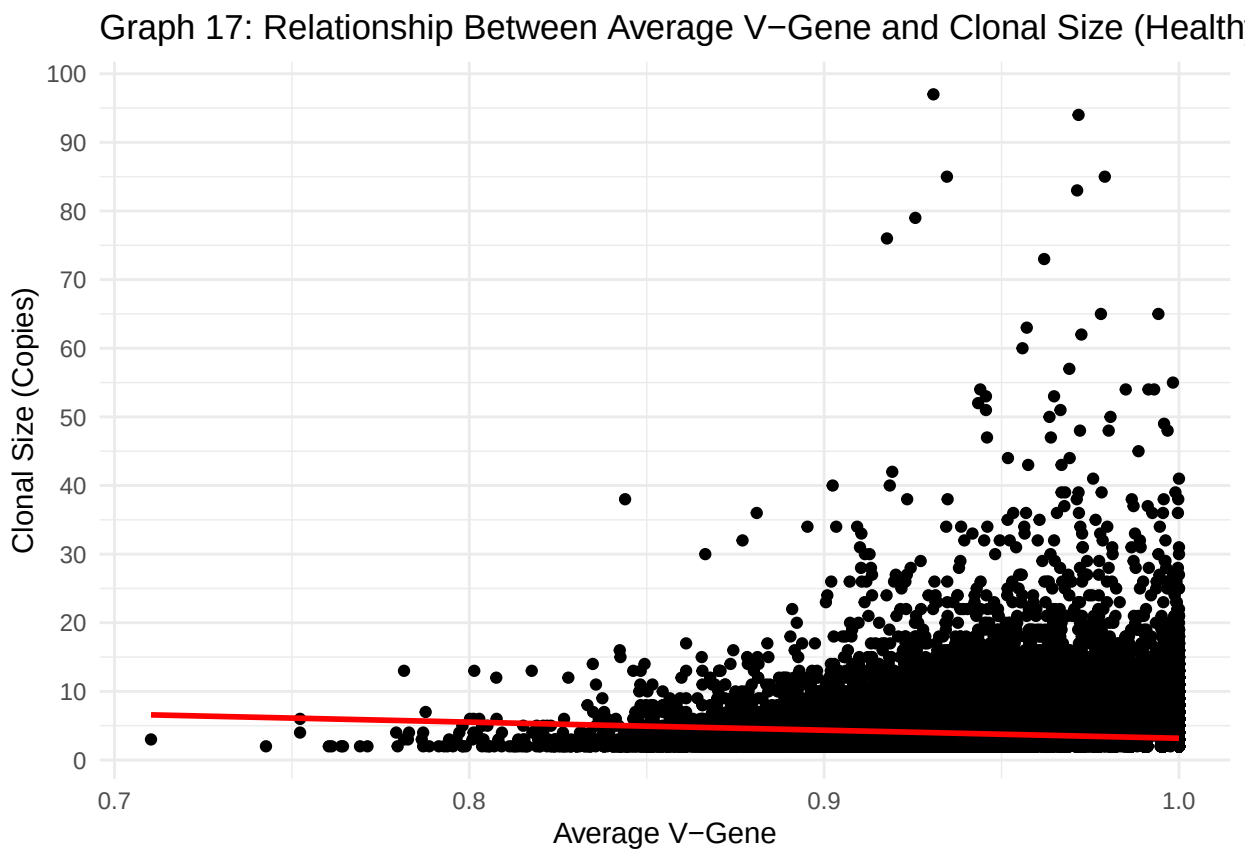
Analysis of Correlation Test (Table 12 - 16) Healthy Group Intercept: 0.9803 Copies Coefficient: -0.001256 (As the clone size increases, the average V-gene identity tends to decrease.) R-squared: 0.01511 Significance: p-value < 2e-16

Diabetes Group Intercept: 0.9749 Copies Coefficient: 0.000388 (As the clone size increases, the average V-gene identity tends to increase.) R-squared: 0.01036 Significance: p-value < 2e-16

Summary Both the healthy and diabetic groups have p-values of 2e-16,indicating strong evidence against the null hypothesis.This suggests that clone size significantly affects average V-gene identity in both groups. In the healthy group, the correlation between average V-gene identity and clone size is negative,while in the diabetic group, the correlation is positive.

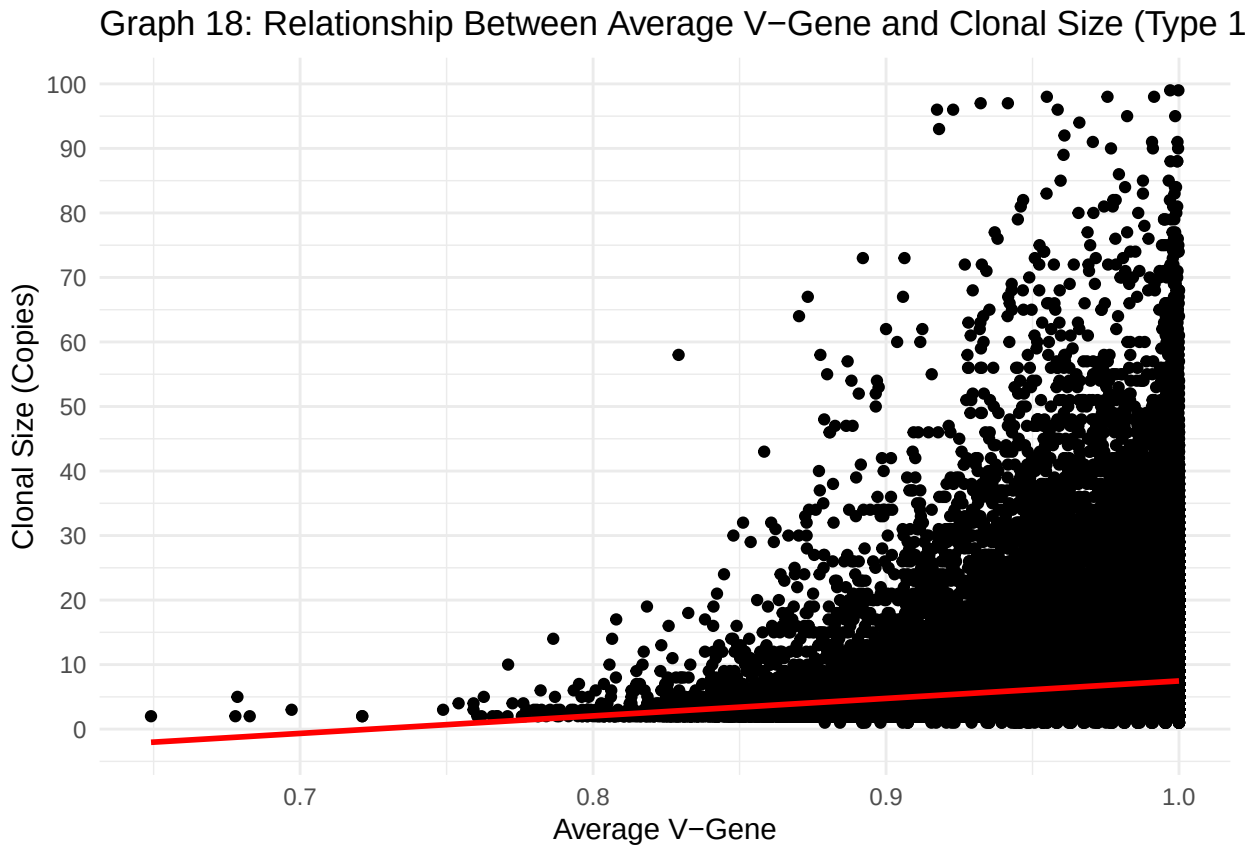
```r
# Data Visualizations
cleaned_data2 <- hpap_cleaned %>% filter(copies < 100) # Remove outliers

# Graph 17: Healthy Group
ggplot(data = cleaned_data2[cleaned_data2$health_status == "Healthy", ], aes(x = avg_v_identity, y = co
  geom_point() +
  geom_smooth(method = "lm", color = "red", se = FALSE) +  # Add regression line based on actual data
  scale_y_continuous(breaks = seq(0, 150, by = 10)) +  # Add custom y-axis breaks
  labs(title = "Graph 17: Relationship Between Average V-Gene and Clonal Size (Healthy)",
       x = "Average V-Gene",
       y = "Clonal Size (Copies)") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
# Graph 18: Diabetes Group
ggplot(data = cleaned_data2[cleaned_data2$health_status == "Diabetes", ], aes(x = avg_v_identity, y = c
  geom_point() +
  geom_smooth(method = "lm", color = "red", se = FALSE) +  # Add regression line based on actual data
  scale_y_continuous(breaks = seq(0, 150, by = 10)) +  # Add custom y-axis breaks
  labs(title = "Graph 18: Relationship Between Average V-Gene and Clonal Size (Type 1 Diabetes)",
       x = "Average V-Gene",
       y = "Clonal Size (Copies)") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Analysis (Table 17 and 18)

Table 17: Healthy Group: As average V-gene identity increases, clonal size shows a slight increase. Most points cluster around an average V-gene identity of 8.5 and a clonal size between 1 and 20.

Table 18: Diabetes Group: As average V-gene identity increases, clonal size increases more dramatically compared to the healthy group. Most points also cluster around an average V-gene identity of 8.5, but clonal size ranges from 1 to 40.

Comparison: The diabetes group demonstrates a more pronounced increase in clonal size compared to the healthy group.