ML_final_project

Integrating host response and unbiased microbe detection for precision sepsis diagnosis in critically ill adults

Introduction

Sepsis is a major threat to public health. It causes nearly 20% of all deaths globally and contributes to one in every two to three hospital deaths in the United States (U.S.). Furthermore, it is the most expensive condition treated in the U.S., costing hospitals more than $38 billion dollars annually [1-3]. Sepsis mortality increases by 8% every hour that appropriate antibiotic therapy is delayed [4], making the rapid and accurate identification of the causative pathogen in bacterial sepsis critical to saving patients' lives. Unfortunately, existing clinical infectious disease diagnostics are largely limited to antiquated, low-yield techniques [5, 6], and therefore, the causative pathogen is rarely identified. As a result, antibiotic treatment often remains empiric rather than pathogen-targeted, with clinical decision-making based on epidemiological information rather than individual patient data [7].

Despite decades of research to improve infection diagnostics, current methods are limited by their reliance on a culturable pathogen, the kinetics of microorganism growth, and clinician-guided testing schema. With the advent of culture-independent methods, these challenges may be overcome. Metagenomic next generation sequencing (mNGS) is an unbiased approach to microbial identification, which can be applied directly to clinical samples [8]. This technique has been utilized for pathogen detection in complex clinical cases posing diagnostic dilemmas due to the failure of traditional diagnostics [9, 10]. mNGS also allows for the identification of host gene expression signatures that distinguish infectious and non-infectious insults [11, 12]. Sepsis, a "life-threatening organ dysfunction due to a dysregulated host response to infection," [13] provides a well-suited opportunity to harness these strengths of mNGS and capture both host and microbe pathophysiology simultaneously.

Methods

Study design, clinical cohort and ethics statement

We conducted a prospective observational study of patients with acute critical illnesses admitted from the ED to the ICU. We studied patients who were enrolled at the University of California, San Francisco (UCSF) and Zuckerberg San Francisco General Hospital. The study was approved by the UCSF Institutional Review Board, which granted a waiver of initial consent for blood sampling. Informed consent was subsequently obtained from patients or their surrogates for continued study participation, as previously described.

For this analysis, inclusion criteria were: (1) admission to the intensive care unit for mechanical ventilation for ARDS or airway protection, (2) age ≥ 18 years, (3) availability of while blood RNA-seq data with 106 protein-coding reads collected within five days of intubation and (4) availability of matched plasma samples to undergo DNA-seq. Exclusion criteria were those patients identified as outliers in the DESeq2 analysis using the variance stabilizing transformation (VST) and principle components analysis (PCA). Subject charts were reviewed by study authors (MA, CL, AL) to confirm a diagnosis of sepsis. Sepsis was adjudicated using the Sepsis-3 definition.

Metagenomic sequencing

Whole blood and plasma were collected into Paxgene and standard tubes, and frozen within 2 hours. To evaluate host gene expression and detect microbes, metagenomic next generation sequencing RNA sequencing (RNA-seq) was performed on whole blood specimens and DNA-seq on plasma. RNA was extracted using the Qiagen RNEasy kit and normalized to 10ng total input per sample. Human cytosolic and mitochondrial ribosomal RNA and globin RNA was depleted using FastSelect (Qiagen). To control for background contamination, we included negative controls (water and HeLa cell RNA) as well as positive controls (spike-in RNA standards from the External RNA Controls Consortium (ERCC). RNA was then fragmented and underwent library preparation using the NEBNext Ultra II RNAseq Kit (New England Biolabs). DNA was extracted from 300uL of plasma using the Zymo Pathogen Magbead Kit before undergoing DNA-seq using the NEBNext Ultra II kit. Libraries underwent 146 nucleotide paired-end Illumina sequencing on an Illumina Novaseq 6000 instrument.

Host differential expression and pathway analysis

Following demultiplexing, sequencing reads were pseudo-aligned with kallisto27 (v. 0.46.1; including bias correction) to an index consisting of all transcripts associated with human protein coding genes (ENSEMBL v. 99), cytosolic and mitochondrial ribosomal RNA sequences, and the sequences of ERCC RNA standards. Gene-level counts were generated from the transcript-level abundance estimates using the R package `txiimport28`, with the `scaledTPM` method.

Differential expression analysis was performed using `DESeq2`. We modeled the expression of individual genes using the design formula `~ Group`. Significant genes were identified using an independent-hypothesis-weighted, Benjamini-Hochberg false discovery rate (FDR) less than 0.1. Empirical Bayesian adaptive shrinkage estimators for $\log_2$-fold change were fit using `ashr`. We generated heatmaps of the top 50 differentially expressed genes by absolute $\log_2$-fold change. visualization, gene expression was normalized using the VST, centered, and z-scaled. Heatmaps were generated using the `pheatmap` package. Unsupervised hierarchical clustering was performed for patients using Euclidean distance and for genes using Manhattan distance. Differentially expressed genes (FDR < 0.05) were analyzed using `Webgestalt`.

Sepsis classifier using random forest

RNA sequencing reads that were used for differential gene expression analyses as outlined above were then used to perform sepsis classification using a random forest model. The dataset with only Group_1 and Group_4 was used with outliers identified by VST transformed gene count heatmaps and PCA analysis included. The dataset was split into training and validation sets at 70%/30%. The `caret` package was used to

perform 10-fold cross-validation to fit the random forest model on the training set. The best random forest model that was identified by cross-validation was then used on the validation set to assess model performance. Model performance was assessed using area under the receiver operating characteristic curve (AUC) using the packages `pROC` and `Epi`. A confusion matrix also presented.

Results

Patient enrollment and sample collection

We conducted a prospective observational study of the EARLI cohort(refs) of critically ill adults admitted from the Emergency Department (ED) to the Intensive Care Unit (ICU). Patients were enrolled at two tertiary care hospitals in San Francisco, California under a research protocol approved by the University of California San Francisco Institutional Review Board (Methods). Following enrollment, whole blood and plasma were collected in the ED and stored at -80C prior to RNA and DNA extraction, respectively. RNA sequencing (RNA-seq) was carried out to assay the host response, and DNA-seq to identify microbes.

Patients with sepsis were clinically adjudicated with blinding to mNGS results using the Sepsis-3 definition [17], and categorized into five subgroups of sepsis status: patients with sepsis due to clinically identified bloodstream infection Group_1 (SepisBldCx+), sepsis due to a microbiologically confirmed primary infection at a site other than the bloodstream Group_2 (SepsisOthCx+), suspected sepsis with negative clinical microbiologic testing Group_3 (Suspected), patients with no evidence of sepsis and a clear alternative explanation for their critical illness Group_4 (No-Sepsis), or patients of indeterminant status Group_5 (Indeterm). Patient demographics are presented in Table 1.

	1_Sepsis+BldCx+	2_Sepsis+OtherCx+	3_Sepsis+Cx-	4_NO_Sepsis	5_Unclear	p	test
n	64	67	68	94	33		
Age (mean (SD))	61.80 (15.33)	67.52 (14.86)	64.37 (15.29)	63.70 (15.46)	70.24 (12.32)	0.049	
APACHEII (mean (SD))	33.05 (10.47)	29.67 (7.33)	32.19 (9.34)	32.05 (9.90)	31.06 (8.22)	0.285	
APACHEIII (mean (SD))	66.42 (22.36)	55.72 (16.34)	64.74 (19.52)	64.26 (18.58)	63.18 (18.83)	0.014	
TempMaxSAPS (mean (SD))	38.07 (1.11)	37.53 (1.43)	37.60 (1.16)	36.92 (1.49)	36.90 (1.43)	<0.001	
TempMinSAPS (mean (SD))	35.46 (2.09)	35.04 (2.10)	35.35 (1.60)	34.49 (2.02)	34.91 (2.10)	0.020	
WBCMaxSAPS (mean (SD))	17.80 (12.13)	15.92 (9.35)	16.11 (9.29)	16.44 (17.29)	12.25 (5.92)	0.352	
WBCMinSAPS (mean (SD))	12.51 (10.46)	12.11 (6.47)	12.01 (7.86)	12.16 (14.66)	10.13 (5.43)	0.870	
HRMaxSAPS (mean (SD))	125.20 (24.33)	109.49 (22.57)	112.19 (24.75)	106.18 (25.64)	103.61 (21.58)	<0.001	
HRMinSAPS (mean (SD))	75.55 (23.71)	66.39 (19.55)	68.24 (20.41)	58.76 (17.78)	61.70 (16.48)	<0.001	
RRMaxSAPS (mean (SD))	35.27 (5.83)	33.00 (8.63)	35.03 (9.24)	34.90 (11.57)	34.48 (10.67)	0.652	
RRMinSAPS (mean (SD))	14.89 (5.95)	12.99 (4.02)	13.60 (5.44)	12.83 (5.73)	12.18 (3.62)	0.074	
SBPMaxSAPS (mean (SD))	155.52 (31.83)	162.30 (32.09)	155.82 (26.33)	167.16 (33.63)	168.52 (29.21)	0.053	
SBPMinSAPS (mean (SD))	70.88 (15.30)	76.96 (16.64)	71.94 (14.22)	76.82 (18.78)	78.45 (16.58)	0.047	
CreatinineMaxSAPS (mean (SD))	2.19 (1.88)	2.12 (2.44)	2.46 (2.92)	2.68 (3.53)	1.50 (1.12)	0.258	
CreatinineMinSAPS (mean (SD))	1.75 (1.42)	1.66 (1.80)	1.91 (2.01)	2.01 (2.35)	1.34 (1.09)	0.453	
PlateletsMinSAPS (mean (SD))	135.72 (115.16)	175.85 (68.11)	173.63 (108.09)	177.30 (80.15)	206.73 (96.79)	0.007	
Gender (%)						0.243	
Male	45 (70.3)	37 (55.2)	38 (55.9)	54 (57.4)	14 (42.4)		
Female	19 (29.7)	29 (43.3)	29 (42.6)	40 (42.6)	19 (57.6)		
Transgender	0 (0.0)	1 (1.5)	1 (1.5)	0 (0.0)	0 (0.0)		
Race (%)						0.443	
Caucasian	22 (34.4)	28 (41.8)	24 (35.3)	37 (39.4)	15 (45.5)		
African American	10 (15.6)	11 (16.4)	12 (17.6)	18 (19.1)	10 (30.3)		
Asian	18 (28.1)	21 (31.3)	23 (33.8)	25 (26.6)	6 (18.2)		
Native American	1 (1.6)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)		
Other	12 (18.8)	5 (7.5)	9 (13.2)	14 (14.9)	2 (6.1)		
Unknown	1 (1.6)	2 (3.0)	0 (0.0)	0 (0.0)	0 (0.0)		
HospWithin30d = 1 (%)	7 (10.9)	7 (10.4)	9 (13.2)	17 (18.1)	8 (24.2)	0.282	
PNA_community = 1 (%)	8 (12.5)	9 (13.4)	16 (23.5)	5 (5.3)	2 (6.1)	0.009	
PNA_healthcare = 1 (%)	7 (10.9)	9 (13.4)	10 (14.7)	2 (2.1)	5 (15.2)	0.041	
COPD2 = 1 (%)	10 (15.6)	15 (22.4)	12 (17.6)	14 (14.9)	14 (42.4)	0.010	

CongestiveHeartFailure = 1 (%)	12 (18.8)	11 (16.4)	16 (23.5)	29 (30.9)	16 (48.5)	0.005
CardiacArrest = 1 (%)	5 (7.8)	7 (10.4)	4 (5.9)	7 (7.4)	3 (9.1)	0.901
PtSepsis = 1 (%)	27 (42.2)	31 (46.3)	32 (47.1)	5 (5.3)	4 (12.1)	<0.001
28d death = 1 (%)	23 (35.9)	14 (20.9)	28 (41.2)	32 (34.0)	12 (36.4)	0.142
60d death = 1 (%)	27 (42.2)	16 (23.9)	31 (45.6)	34 (36.2)	13 (39.4)	0.094
Hospital death = 1 (%)	24 (37.5)	15 (22.4)	28 (41.2)	30 (31.9)	11 (33.3)	0.192
MechVent = 1 (%)	49 (76.6)	59 (88.1)	60 (88.2)	90 (95.7)	31 (93.9)	0.005
SIRS_HR = 1 (%)	58 (90.6)	53 (79.1)	57 (83.8)	70 (74.5)	22 (66.7)	0.032
SIRS_temp = 1 (%)	51 (79.7)	51 (76.1)	54 (79.4)	71 (75.5)	24 (72.7)	0.915
SIRS_RR = 1 (%)	64 (100.0)	64 (95.5)	68 (100.0)	92 (97.9)	33 (100.0)	0.157
SIRS_WBC = 1 (%)	53 (82.8)	45 (67.2)	48 (70.6)	54 (57.4)	18 (54.5)	0.008
SIRS_total (%)						0.014
1	0 (0.0)	1 (1.5)	0 (0.0)	1 (1.1)	1 (3.0)	
2	6 (9.4)	11 (16.4)	11 (16.2)	22 (23.4)	9 (27.3)	
3	18 (28.1)	30 (44.8)	23 (33.8)	42 (44.7)	14 (42.4)	
4	40 (62.5)	25 (37.3)	34 (50.0)	29 (30.9)	9 (27.3)	
Intubated = 1 (%)	49 (76.6)	58 (86.6)	62 (91.2)	87 (92.6)	31 (93.9)	0.020
OnPressorsSAPS = 1 (%)	57 (89.1)	49 (73.1)	60 (88.2)	61 (64.9)	21 (63.6)	<0.001
virusPresent = 1 (%)	5 (7.8)	22 (32.8)	0 (0.0)	0 (0.0)	0 (0.0)	<0.001
Immunocompromised = 1 (%)	8 (12.5)	5 (7.5)	11 (16.2)	6 (6.4)	6 (18.2)	0.157
SOT = 1 (%)	3 (4.7)	0 (0.0)	3 (4.4)	1 (1.1)	1 (3.0)	0.296
HTN = 1 (%)	16 (25.0)	31 (46.3)	31 (45.6)	44 (46.8)	12 (36.4)	0.045
Cirrhosis = 1 (%)	8 (12.5)	4 (6.0)	10 (14.7)	6 (6.4)	2 (6.1)	0.247
CKD = 1 (%)	12 (18.8)	15 (22.4)	15 (22.1)	29 (30.9)	7 (21.2)	0.438
Malignancy = 1 (%)	13 (20.3)	14 (20.9)	14 (20.6)	20 (21.3)	7 (21.2)	1.000
HIV = 1 (%)	7 (10.9)	4 (6.0)	3 (4.4)	7 (7.4)	0 (0.0)	0.281
Diabetes = 1 (%)	18 (28.1)	22 (32.8)	17 (25.0)	31 (33.0)	13 (39.4)	0.599

Host signature of blood culture positive sepsis

We assessed transcriptional differences between patients with sepsis due to bloodstream infection Group_1 (SepisBldCx+) versus those with no evidence of infection Group_4 (No-Sepsis). More than 5000 differentially expressed genes were identified at an adjusted p value (padj) < 0.05 and unsupervised hierarchical clustering revealed clear separation of groups (Figure 1) using the top 500 differentially expressed genes. Gene set enrichment analysis (GSEA) demonstrated upregulation in pathways related to cytokine signaling and innate immune defense, consistent with acute infection (Figure 2).

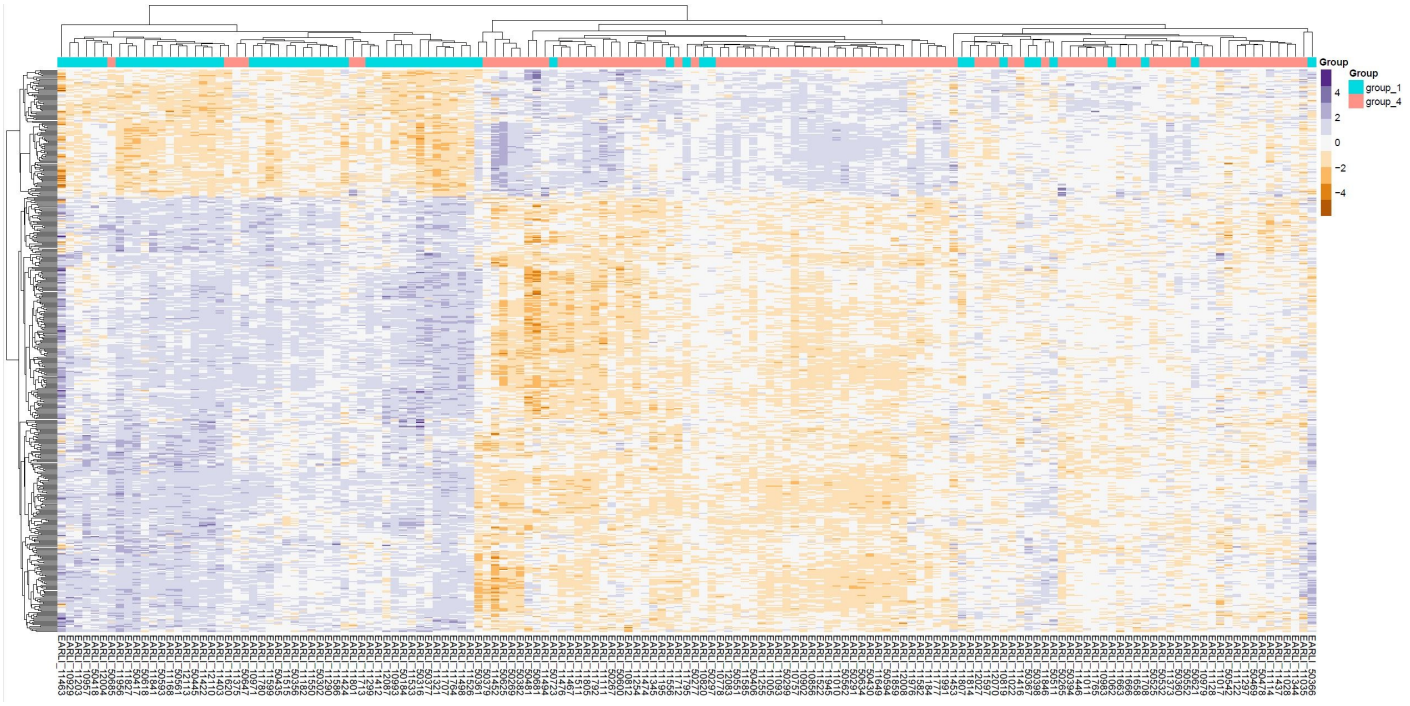


Figure 1

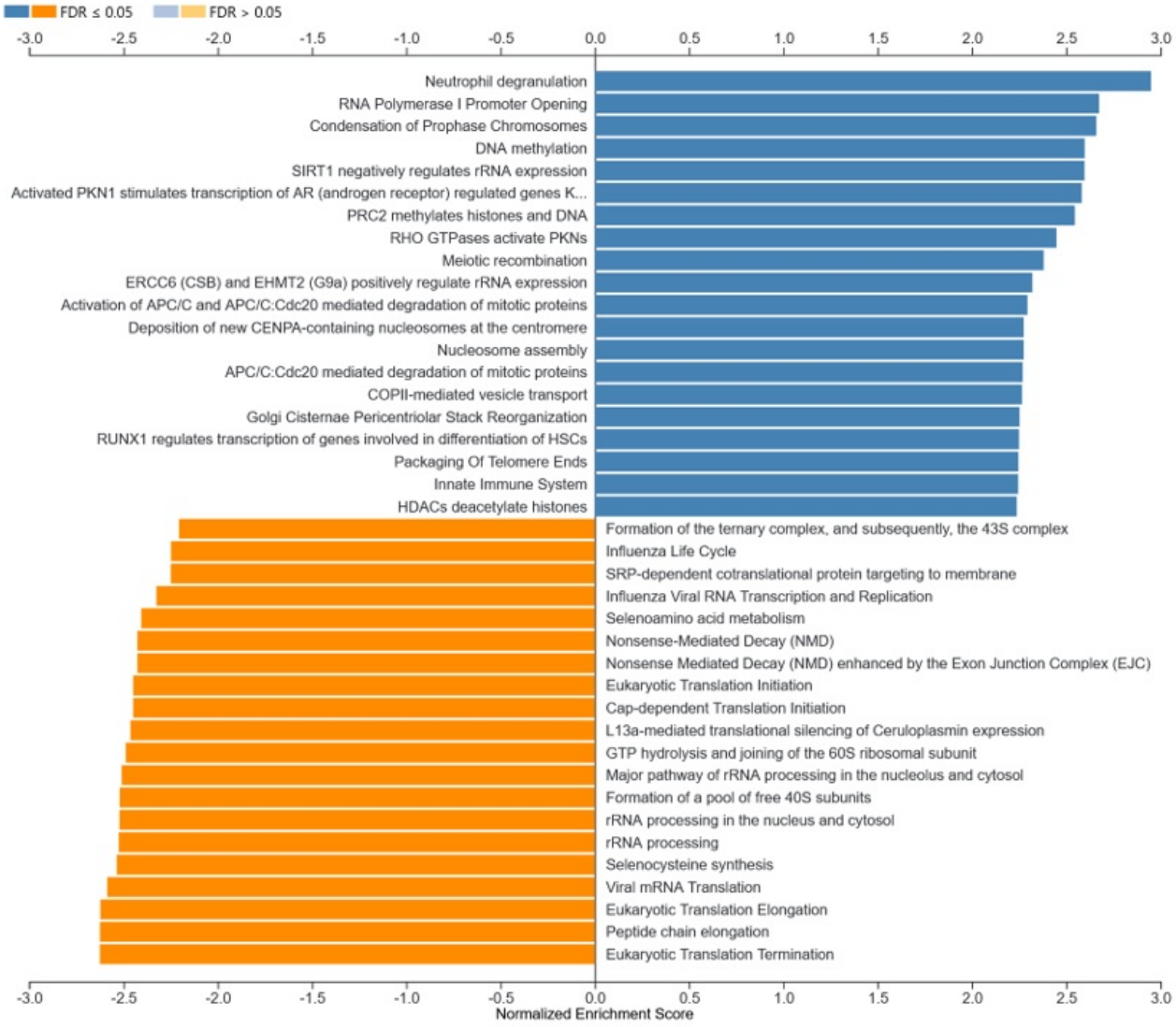


Figure 2

Host transcriptional classifier for bloodstream infection-related sepsis

After characterizing the biological pathways represented in the sepsis host transcriptional signature, we next sought to leverage this signature for clinical diagnosis. DE analysis suggested that BSI were characterized by greater uniformity of host expression compared to sepsis originating from diverse peripheral anatomical location, so we first focused on distinguishing Group_1 (SepisBldCx+) and Group_4 (No-Sepsis). After dividing the cohort into an independent training and validation set using an 70%/30% split, we employed a random forest model. 10-fold cross-validation was performed on the training set and model performance was assessed using a confusion matrix and area under the receiver operating characteristic

curve (AUC).

This yielded a BSI classifier that performed with an accuracy of 0.93 (95% CI 0.82 - 0.99) (Table 2) and an AUC of 0.92 (95% CI 0.83 – 1.0) (Figure 3). The confidence interval for the AUC was calculated using 2000 boostrap samples.

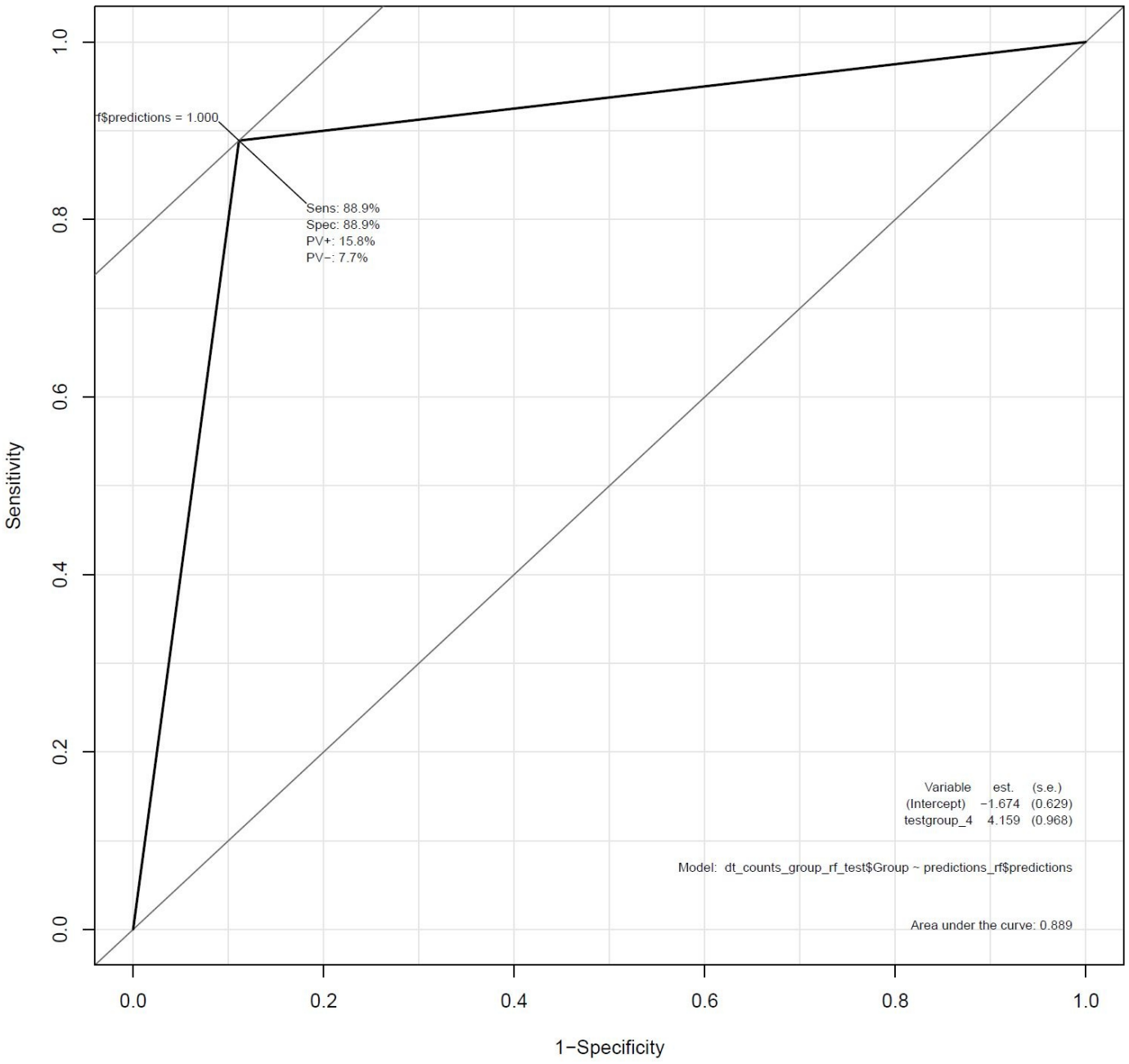


Figure 3

Table 2a - Confusion matrix: bolded labels are the true class

	group_1	**group_4**
group_1	15	0
group_4	3	27

Table 2b - Accuracy, kappa value, and 95% confidence interval of accuracy

	x
Accuracy	0.9333333
Kappa	0.8571429
AccuracyLower	0.8173155
AccuracyUpper	0.9860349

Table 2c - Sensitivity, specificity, positive predictive value, and negative predictive value

	x
Sensitivity	0.8333333
Specificity	1.0000000

Pos Pred Value	1.0000000
Neg Pred Value	0.9000000

Discussion

Sepsis is defined as a dysregulated host response to infection yet no existing clinical diagnostics evaluate both critical aspects of the disease. Here we present a novel approach to integrating host and microbial metrics using metagenomic sequencing. We demonstrate that this approach identified important differences in gene expression between bacteremic sepsis patients and critically ill patients without sepsis. Unsupervised hierarchical clustering was able to distinguish these two patient groups. Further, a random forest model was able to distinguish between bacteremic sepsis and critical illness without sepsis at an AUC of 0.90. These results set the stage for a transformation in sepsis diagnostics. Metagenomic sequencing can be performed at the bedside in order to help clinicians decide on triage and adminsitration of antibiotics. Next steps involve integrating the microbial information from metagenomic seqencing into the host response data presented here.

References

1. Rudd, K.E., et al., Global, regional, and national sepsis incidence and mortality, 1990-2017: analysis for the Global Burden of Disease Study. Lancet, 2020. 395(10219): p. 200-211.
2. Liu, V., et al., Hospital deaths in patients with sepsis from 2 independent cohorts. JAMA, 2014. 312(1): p. 90-2.
3. Liang, L.A., Moore B (IBM Watson Health), Soni A (AHRQ). National Inpatient Hospital Costs: The Most Expensive Conditions by Payer, 2017. HCUP Statistical Brief #261 2020 7/2/2020 [cited 2020 Sept. 2]; Available from: www.hcup-us.ahrq.gov/reports/statbriefs/sb261-Most-Expensive-Hospital-Conditions-2017.pdf.
4. Ferrer, R., et al., Empiric antibiotic treatment reduces mortality in severe sepsis and septic shock from the first hour: results from a guideline-based performance improvement program. Crit Care Med, 2014. 42(8): p. 1749-55.
5. Lamy, B., et al., What is the relevance of obtaining multiple blood samples for culture? A comprehensive model to optimize the strategy for diagnosing bacteremia. Clin Infect Dis, 2002. 35(7): p. 842-50.
6. Aronson, M.D. and D.H. Bor, Blood cultures. Ann Intern Med, 1987. 106(2): p. 246-53.
7. Timbrook, T.T., et al., The Effect of Molecular Rapid Diagnostic Testing on Clinical Outcomes in Bloodstream Infections: A Systematic Review and Meta-analysis. Clin Infect Dis, 2017. 64(1): p. 15-23.
8. Bibby, K., Metagenomic identification of viral pathogens. Trends Biotechnol, 2013. 31(5): p. 275-9.
9. Wilson, M.R., et al., Actionable diagnosis of neuroleptospirosis by next-generation sequencing. N Engl J Med, 2014. 370(25): p. 2408-17.
10. Wilson, M.R., et al., Diagnosing Balamuthia mandrillaris Encephalitis With Metagenomic Deep Sequencing. Ann Neurol, 2015. 78(5): p. 722-30.
11. Sweeney, T.E., H.R. Wong, and P. Khatri, Robust classification of bacterial and viral infections via integrated host gene expression diagnostics. Sci Transl Med, 2016. 8(346): p. 346ra91.
12. Tsalik, E.L., et al., Host gene expression classifiers diagnose acute respiratory illness etiology. Sci Transl Med, 2016. 8(322): p. 322ra11.
13. Singer, M., et al., The Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3). JAMA, 2016. 315(8): p. 801-10.

Code for reproduction of analysis

Data preparation

Read in the RNA seq counts and metadata and modify for analysis

```
# read in the raw RNA seq counts
raw_counts <- fread("filtered_genecounts.csv")

# update the column name V1 -> ensg_name
setnames(raw_counts, old = "V1", new = "ensg_name")

# EARLI_{barcode} is the naming scheme for the raw_counts columns; metadata file has row names that are EARLI stu
dy ID
metadata <- fread("final_metadata_used_in_Table1_wBarcode.csv")

# remove the V1 column
metadata[,V1 := NULL]

# create a new column that will be EARLI_{barcode} in metadata file to mirror the raw_counts file
metadata_rownames <- paste0("EARLI_", metadata$barcode)
metadata[, row_names := metadata_rownames]
setcolorder(metadata, neworder = "row_names")

# filter the metadata to contain only EARLI_{barcode}s that are also in the raw_counts data; down to 319 subjects
metadata_filtered <- metadata[row_names %in% colnames(raw_counts)]

# remove rows not needed
metadata_filtered[, `:=`(EARLIStudyId=NULL, ER_admit_date=NULL, BirthDate=NULL)]
```

Find genes that are in at least 30% of the samples

```
# generate a matrix of the raw counts without the first row (gene names)
matrix_raw_counts <- as.matrix(raw_counts[,-c("ensg_name")])

# name the rows of the matrix using the gene name column
rownames(matrix_raw_counts) <- raw_counts$ensg_name

# create a matrix of TRUE if raw_count >=1, FALSE if raw_count 0
present_matrix <- matrix_raw_counts >= 1

# get a logical index that is TRUE if the sum of the ROWS (i.e., by gene) is greater than 30% of the total number
of COLUMNS (i.e., number of samples) and false if it is <=30% of the samples
filter_idx <- apply(present_matrix, 1, sum) > ncol(matrix_raw_counts)*.3

# get a final matrix of genes only present >30% of the time by using the filter index by row
matrix_counts_30plus <- matrix_raw_counts[filter_idx == "TRUE",]

dim(matrix_raw_counts)
```

```
## [1] 27097   319
```

```
dim(matrix_counts_30plus)
```

```
## [1] 25953   319
```

Prepare the data for entry into the DESeq() function

```
# make the counts matrix a data.frame
df_counts_30plus <- as.data.frame(matrix_counts_30plus)

# (1) make the metadata a data frame, (2) set the row names, and (3) delete the row_names and EARLI study ID colu
mns
df_metadata_filtered <- as.data.frame(metadata_filtered)
rownames(df_metadata_filtered) <- df_metadata_filtered$row_names
df_metadata_filtered$row_names <- NULL

# change the name of group factor so it plays well with DESeq2
df_metadata_filtered$Group <- factor(df_metadata_filtered$Group,
                                     levels = c("1_Sepsis+BldCx+",
                                                "2_Sepsis+OtherCx+",
                                                "3_Sepsis+Cx-", "4_NO_Sepsis",
                                                "5_Unclear"),
                                     labels = c("group_1", "group_2", "group_3",
                                                "group_4", "group_5"))

# subset the groups to include only those in group 1, 2, and 4
df_metadata_filtered_g124 <-
    df_metadata_filtered[df_metadata_filtered$Group %in%
                             c("group_1", "group_2", "group_4"),]
df_metadata_filtered_g124$Group <- droplevels(df_metadata_filtered_g124$Group)
#* down to 220 subjects

# ensure that the COLUMNS of the RAW counts data are in the same order as the ROWS of the METADATA
matching_idx <- match(rownames(df_metadata_filtered_g124),
                      colnames(df_counts_30plus))

# sort using matching_idx
df_counts_30plus <- df_counts_30plus[, matching_idx]

# check that the rows of the metadata are the same as the columns of the count data
all(rownames(df_metadata_filtered_g124) == colnames(df_counts_30plus))
```

```
## [1] TRUE
```

```
# remove unnecessary data from global environment
rm("filter_idx", "matching_idx", "matrix_counts_30plus", "matrix_raw_counts",
   "metadata", "metadata_filtered", "metadata_rownames", "present_matrix",
   "raw_counts")
```

Random forest model

Prepare the data

```
# set the seed
set.seed(20211003)

# drop group 2 from the metadata
df_metadata_filtered_g14 <-
  df_metadata_filtered_g124[df_metadata_filtered_g124$Group %in%
                              c("group_1", "group_4"),]
df_metadata_filtered_g14$Group <- droplevels(df_metadata_filtered_g14$Group)

# match the gene counts
matching_idx_rf <- match(rownames(df_metadata_filtered_g14),
                         colnames(df_counts_30plus))

# sort using matching_idx_rf
df_counts_30plus_rf <- df_counts_30plus[, matching_idx_rf]

# transpose the data for running the model
t_df_counts_30plus_rf <- t(df_counts_30plus_rf)

# create a dataset to merge
df_merge_rf <- df_metadata_filtered_g14[, c("Group"), drop=F]
dt_merge_rf <- data.table(df_merge_rf, keep.rownames = TRUE)
setnames(dt_merge_rf, "rn", "EARLI_ID")

dt_t_counts_30plus_rf <- data.table(t_df_counts_30plus_rf, keep.rownames = TRUE)
setnames(dt_t_counts_30plus_rf, "rn", "EARLI_ID")

dt_counts_group_rf <- merge(dt_merge_rf, dt_t_counts_30plus_rf, by="EARLI_ID")
```

Split the data into training and validation

```
# set the seed
set.seed(20211003)

# create the sampling index
sampler <- createDataPartition(dt_counts_group_rf$Group,
                               times = 1,
                               p = 0.7,
                               list = FALSE)

# training set
dt_counts_group_rf_train <- dt_counts_group_rf[sampler, ]
nrow(dt_counts_group_rf_train)
```

```
## [1] 109
```

```
# test set
dt_counts_group_rf_test <- dt_counts_group_rf[-sampler, ]
nrow(dt_counts_group_rf_test)
```

```
## [1] 45
```

Perform cross-validation of random forest model

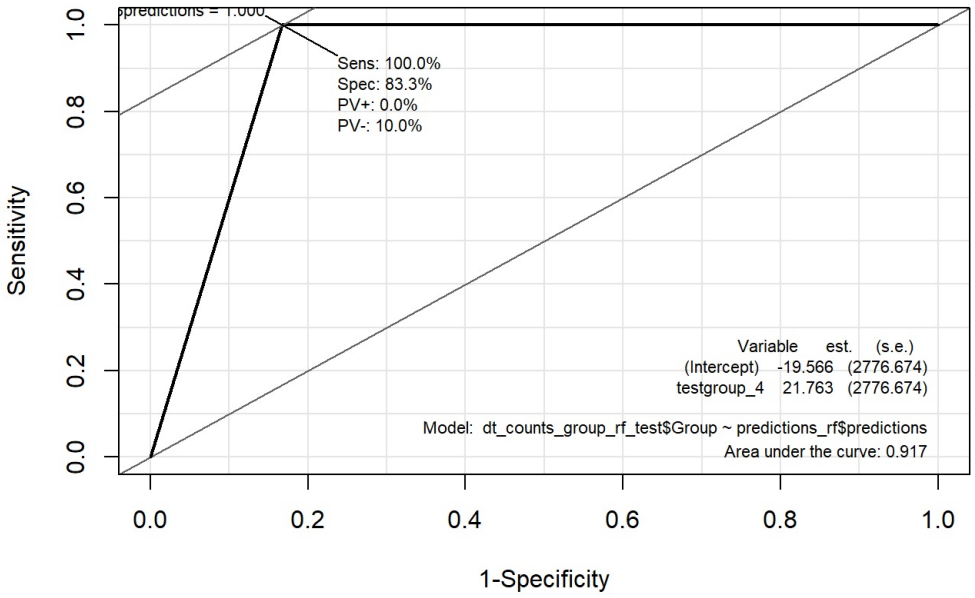
```
# set the seed
set.seed(20211003)

# perform 3-fold cross-validation
fitControl <- trainControl(method = "cv", number = 10)

# set the range of mtry values to tune across
mtry <- c(120, 130, 140, 150, 160, 170, 180, 190, 200)

# set the splitrule values to tune across
splitrule <- c("gini", "extratrees", "hellinger")

# set the min.node.size to tune across
min.node.size <- c(1,2,3,4,5)

# generate a matrix of C and sigma values to use for hyperparameter tuning
tune.matrix <- expand.grid(mtry = mtry, splitrule = splitrule,
                           min.node.size = min.node.size)

# tune the SVM model using 3-fold cross-validation and the tune.matrix above
cv.rf.model <- train(x = dt_counts_group_rf_train[,-c(1,2)],
                     y = dt_counts_group_rf_train$Group,
                     trControl = fitControl,
                     method = "ranger",
                     tuneGrid = tune.matrix)

cv.rf.model$bestTune
```

```
##     mtry splitrule min.node.size
## 104  180 hellinger             4
```

```
tuned.rf.model <- cv.rf.model$finalModel
```

Predict on the test dataset

```
# set the seed
set.seed(20211003)

predictions_rf <- predict(tuned.rf.model,
                          data = dt_counts_group_rf_test,
                          type = "response")

# Look at the confusion matrix
conf_matrix <- confusionMatrix(predictions_rf$predictions,
                               dt_counts_group_rf_test$Group)

# Assess the ROC curve
ROC(test = predictions_rf$predictions,
    stat = dt_counts_group_rf_test$Group,
    plot = "ROC")
```




```
# make factors appropriate for roc()
true_values <- factor(dt_counts_group_rf_test$Group,
                      levels = c("group_4", "group_1"),
                      ordered=T)

rf_preds <- factor(predictions_rf$predictions,
                   levels = c("group_4", "group_1"),
                   ordered=T)

# get ROC confidence interval
proc_obj <- roc(response = true_values, predictor = rf_preds, ci = TRUE)
```

```
## Setting levels: control = group_4, case = group_1
```

```
## Warning in value[[3L]](cond): Ordered predictor converted to numeric vector.
## Threshold values will not correspond to values in predictor.
```

```
## Setting direction: controls < cases
```

```
proc_obj
```

```
##
## Call:
## roc.default(response = true_values, predictor = rf_preds, ci = TRUE)
##
## Data: rf_preds in 27 controls (true_values group_4) < 18 cases (true_values group_1).
## Area under the curve: 0.9167
## 95% CI: 0.8281-1 (DeLong)
```

Start DESeq workflow

Create the DESeq object

```
dds_earli_groups <- DESeqDataSetFromMatrix(countData = df_counts_30plus,
                                           colData = df_metadata_filtered_g124,
                                           design = ~ Group)
```

Rewrite plotPCA function to plot PCAs 3 and 4

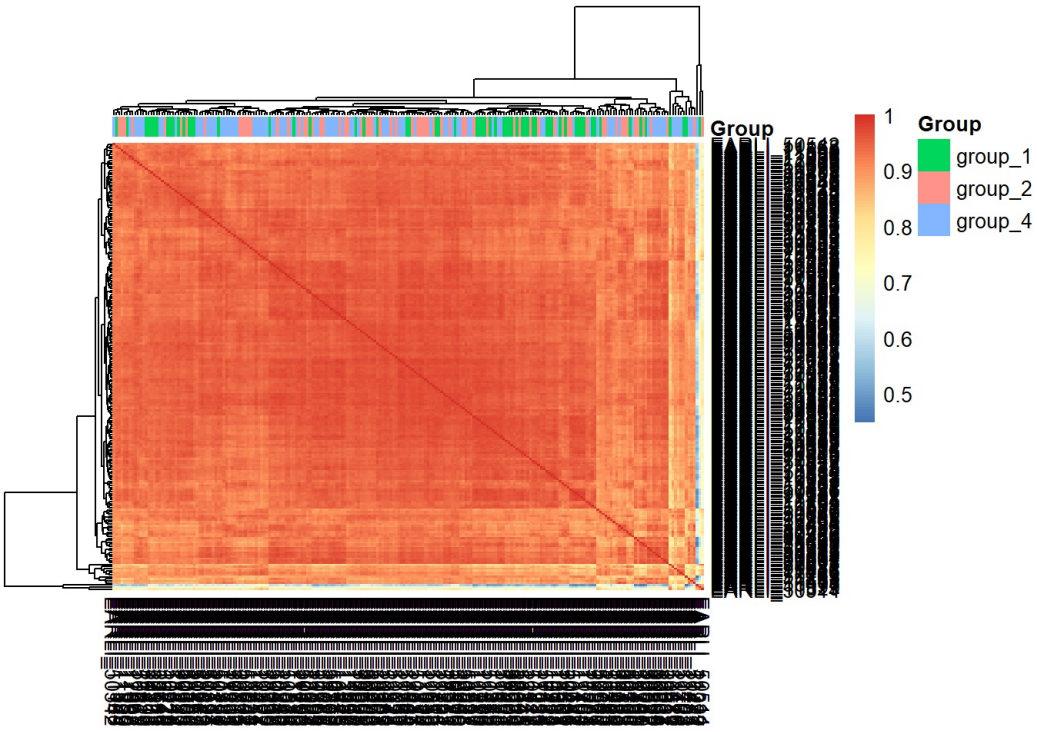
```
#* rewrite the plotPCA function to plot the 3rd and 4th PCs
plotPCA34 <- function (object, intgroup = "condition", ntop = 500, returnData = FALSE)
{
    rv <- rowVars(assay(object))
    select <- order(rv, decreasing = TRUE)[seq_len(min(ntop,
                                                       length(rv)))]
    pca <- prcomp(t(assay(object)[select, ]))
    percentVar <- pca$sdev^2/sum(pca$sdev^2)
    if (!all(intgroup %in% names(colData(object)))) {
        stop("the argument 'intgroup' should specify columns of colData(dds)")
    }
    intgroup.df <- as.data.frame(colData(object)[, intgroup,
                                                 drop = FALSE])
    group <- if (length(intgroup) > 1) {
        factor(apply(intgroup.df, 1, paste, collapse = ":"))
    } else {
        colData(object)[[intgroup]]
    }

    d <- data.frame(PC3 = pca$x[, 3], PC4 = pca$x[, 4], group = group,
                    intgroup.df, name = colnames(object))

    if (returnData) {
        attr(d, "percentVar") <- percentVar[3:4]
        return(d)
    }

    ggplot(data = d, aes_string(x = "PC3", y = "PC4",color = "group")) +
        geom_point(size = 3) +
        xlab(paste0("PC3: ", round(percentVar[3] * 100), "% variance")) +
        ylab(paste0("PC4: ", round(percentVar[4] * 100), "% variance")) +
        coord_fixed()
}
```

Quality assessment metrics

```
# normalize the raw counts
dds_earli_groups <- estimateSizeFactors(dds_earli_groups)

# log transform the data
vst_earli_groups <- vst(dds_earli_groups, blind = TRUE)
vst_matrix <- assay(vst_earli_groups)
vst_correlations <- cor(vst_matrix)
```

Generate the heatmap to assess for outliers

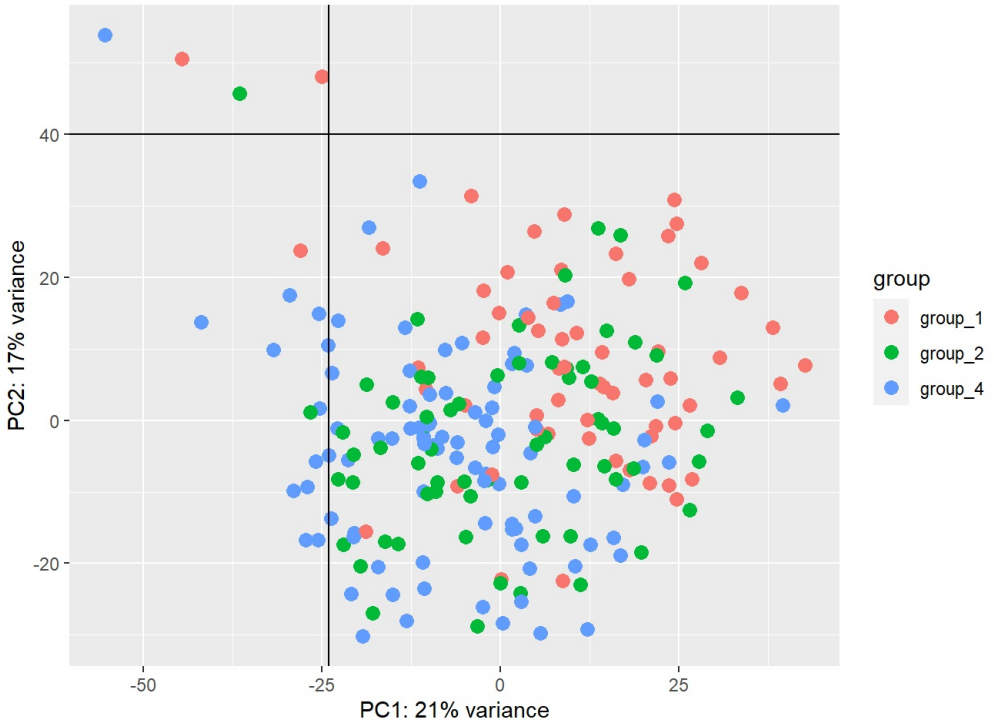
```
pheatmap(vst_correlations,
         annotation_col = df_metadata_filtered_g124[,"Group", drop = FALSE])
```



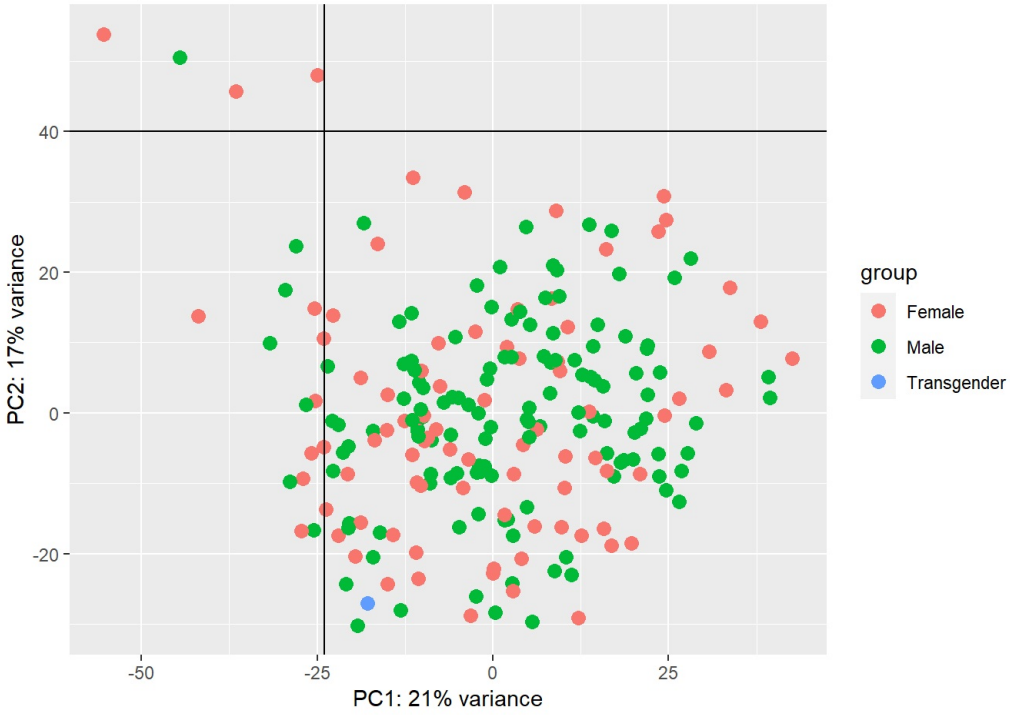
Significant outliers on the heatmap: EARLI_11997, EARLI_11329, EARLI_50514, EARLI_11417. which are noted by the significantly different variances from the normalized counts created in the `vst_correlations` object.

Perform PCA to assess for outliers

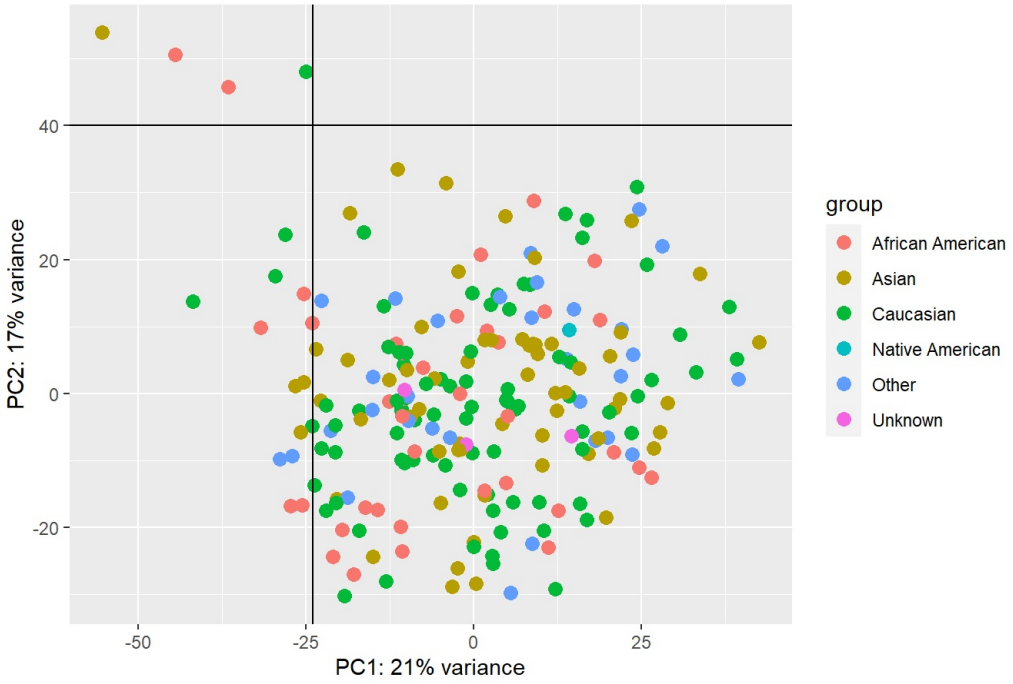
```
plotPCA(vst_earli_groups, intgroup = "Group", ntop = 1000)+
   geom_vline(xintercept = -24) + geom_hline(yintercept = 40)
```



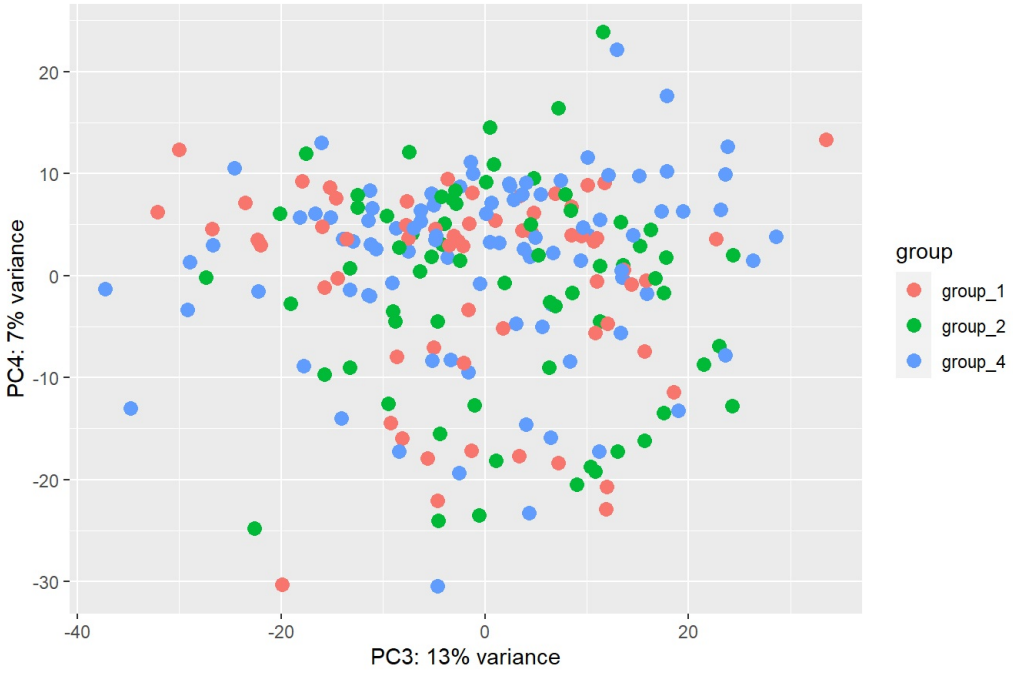

```
plotPCA(vst_earli_groups, intgroup = "Gender", ntop = 1000)+
   geom_vline(xintercept = -24) + geom_hline(yintercept = 40)
```



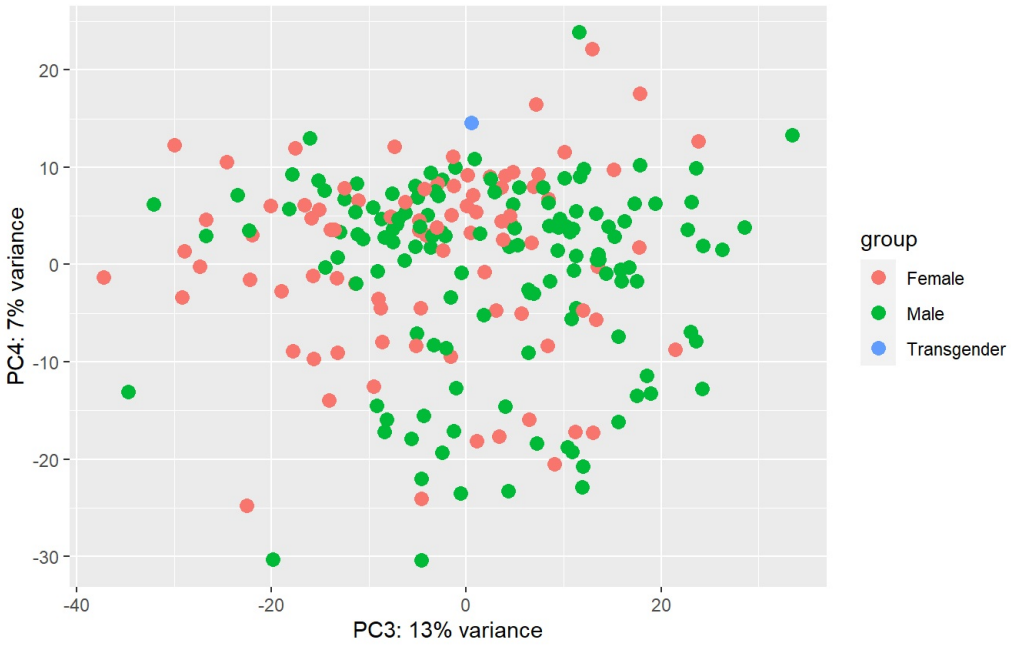

```
plotPCA(vst_earli_groups, intgroup = "Race", ntop = 1000)+
   geom_vline(xintercept = -24) + geom_hline(yintercept = 40)
```



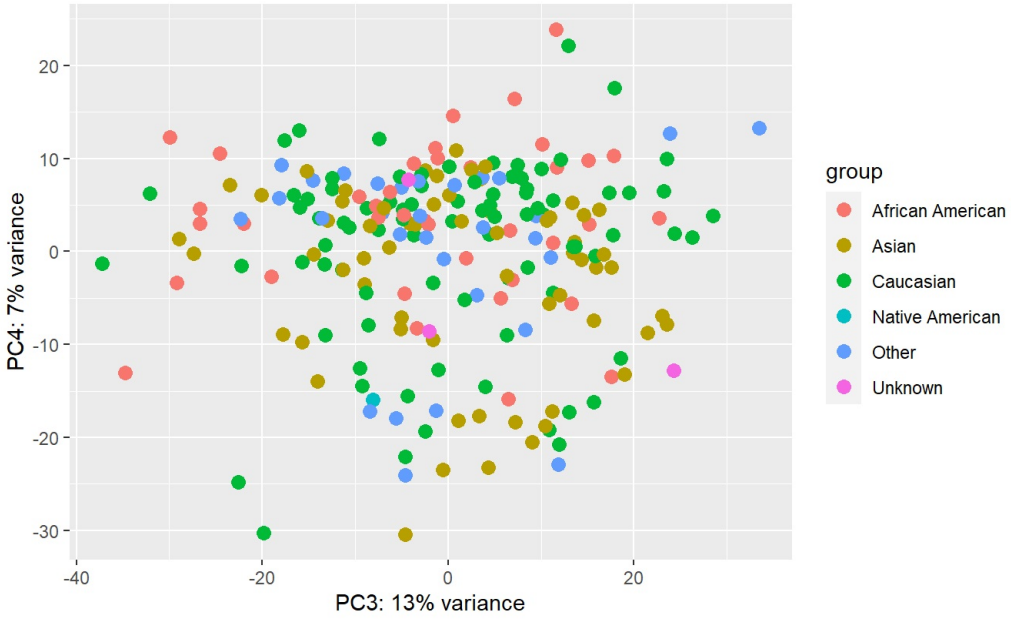

```
plotPCA34(vst_earli_groups, intgroup = "Group", ntop = 1000)
```



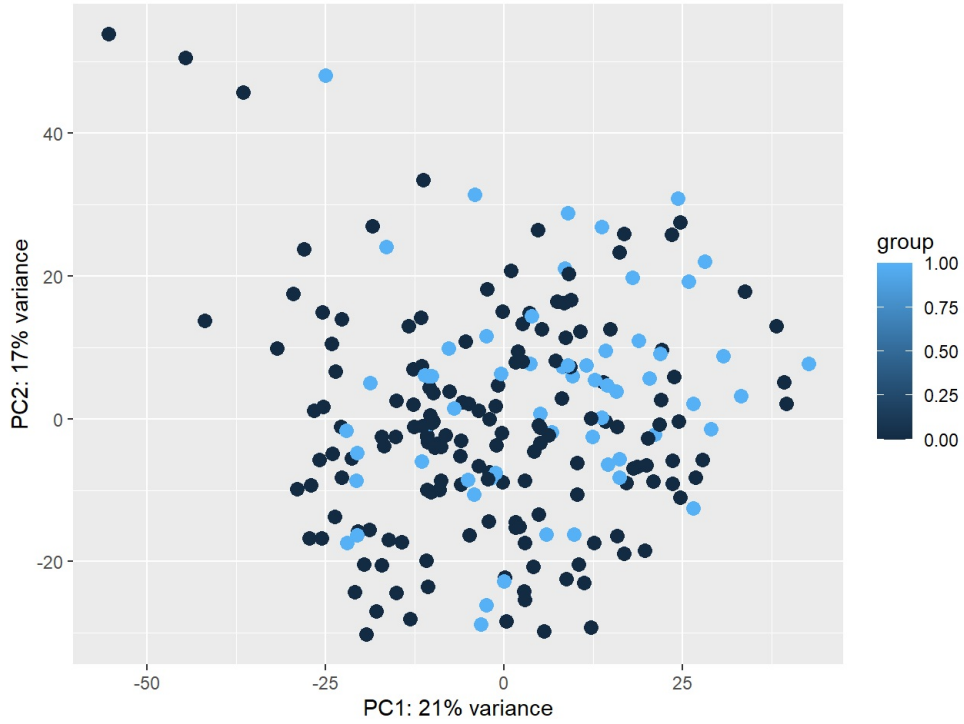

```
plotPCA34(vst_earli_groups, intgroup = "Gender", ntop = 1000)
```



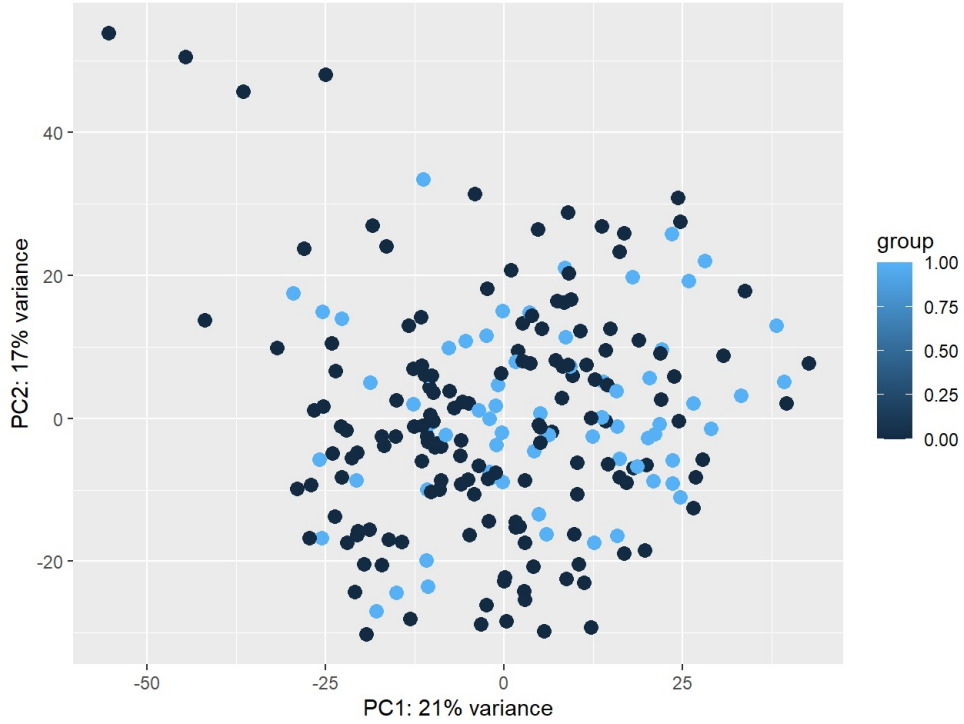

```
plotPCA34(vst_earli_groups, intgroup = "Race", ntop = 1000)
```



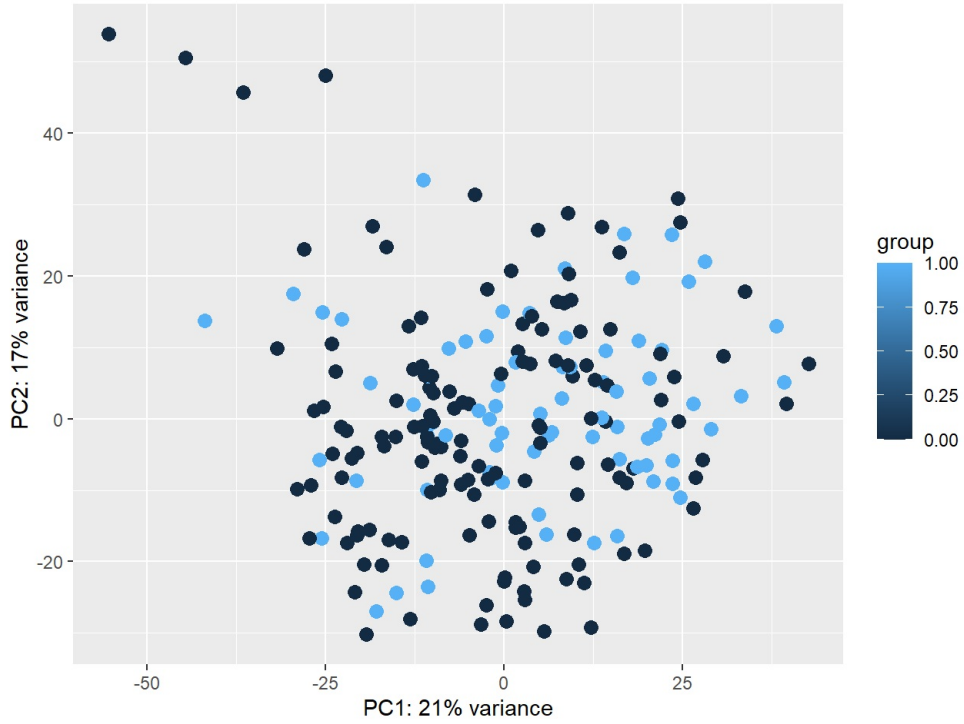

```
# plot other possible explanatory variables
plotPCA(vst_earli_groups, intgroup="PtSepsis", ntop=1000)
```




```
plotPCA(vst_earli_groups, intgroup="28d death", ntop=1000)
```




```
plotPCA(vst_earli_groups, intgroup="60d death", ntop=1000)
```



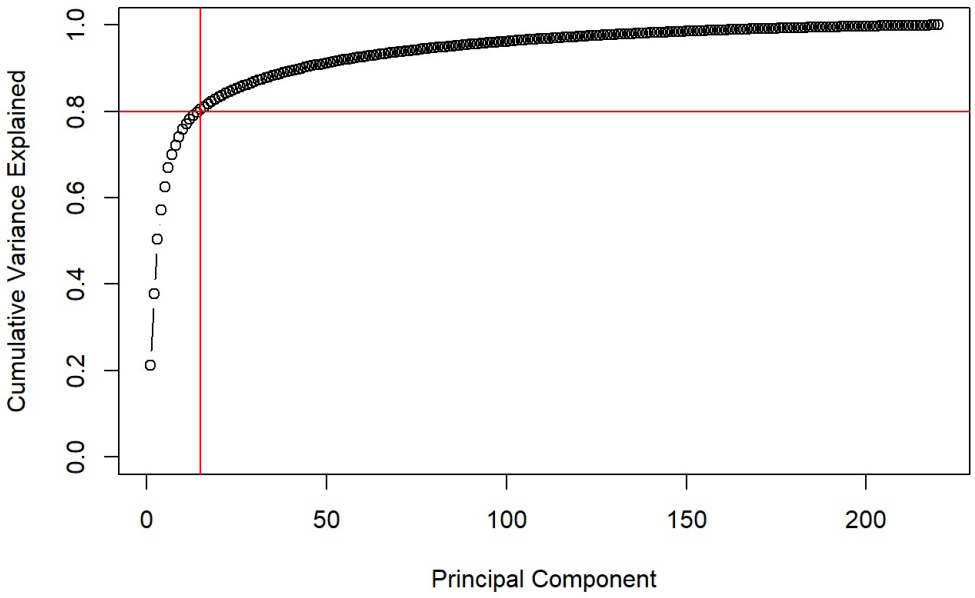
The PCA plots are used to assess whether major explanatory variables are associated with separation by the most significantly explanatory principle components. There are four outlying points noted in the plots above that are likely outliers. These need to be identified and compared to those outlying points in the heatmap above.

Perform manual PCA

```
# get the PCA object for plotting
row_variance <- rowVars(vst_matrix)
select_idx <- order(row_variance, decreasing = TRUE)[1:1000]
transposed_selected_vst_matrix <- t(vst_matrix[select_idx,])
pca_values <- prcomp(transposed_selected_vst_matrix)

# get the percentage of variance explained by each PC
percent_var_explained <- pca_values$sdev^2/sum((pca_values$sdev)^2)

# minimum number of PCs explaining 80% of the variance
(minPCs <- min(which(cumsum(percent_var_explained) > 0.8)))
```

```
## [1] 15
```

```
# plot cumulative variance explained
plot(1:length(percent_var_explained),
     cumsum(percent_var_explained),
     xlab="Principal Component",
     ylab="Cumulative Variance Explained", ylim=c(0,1), type='b')
abline(h=0.8, col="red")
abline(v=15, col="red")
```



Find the outliers in the PCA analysis

```
# which EARLI_{barcode}s have PC1 < -24 and PC2 > 40?
pca1andpca2 <- pca_values$x[,1:2]
outliers_idx <- which(pca1andpca2[,1] < -24 & pca1andpca2[,2] > 40)
pca1andpca2[outliers_idx,]
```

```
##                   PC1      PC2
## EARLI_11329 -44.54455 50.53825
## EARLI_11417 -24.97368 48.02750
## EARLI_50514 -36.53306 45.69447
## EARLI_11997 -55.35811 53.87412
```

These outliers in the PCA are also outliers in the heatmap.

EARLI_barcode	PC1	PC2
EARLI_11329	-44.54455	50.53825
EARLI_11417	-24.97368	48.02750
EARLI_50514	-36.53306	45.69447
EARLI_11997	-55.35811	53.87412

Remove the identified outliers

```
# pull the names of the outliers
names_outliers <- rownames(pca1andpca2[outliers_idx,])

# remove outliers prior to starting the final DE analysis
df_counts_30plus_noOutliers <-
    df_counts_30plus[!(colnames(df_counts_30plus) %in% names_outliers)]
df_metadata_filtered_g124_noOutliers <-
    df_metadata_filtered_g124[!(rownames(df_metadata_filtered_g124) %in% names_outliers),]

# remove unnecessary objects from global environment
rm("vst_correlations","vst_earli_groups","vst_matrix","select_idx",
"transposed_selected_vst_matrix","row_variance","pca1andpca2",
"percent_var_explained","plotPCA34","names_outliers","outliers_idx",
"p1","p2","p3","p4","p5","p6","pca_values","dds_earli_groups","df_counts_30plus",
"df_metadata_filtered", "df_metadata_filtered_g124", "minPCs")
```

```
## Warning in rm("vst_correlations", "vst_earli_groups", "vst_matrix",
## "select_idx", : object 'p1' not found
```

```
## Warning in rm("vst_correlations", "vst_earli_groups", "vst_matrix",
## "select_idx", : object 'p2' not found
```

```
## Warning in rm("vst_correlations", "vst_earli_groups", "vst_matrix",
## "select_idx", : object 'p3' not found
```

```
## Warning in rm("vst_correlations", "vst_earli_groups", "vst_matrix",
## "select_idx", : object 'p4' not found
```

```
## Warning in rm("vst_correlations", "vst_earli_groups", "vst_matrix",
## "select_idx", : object 'p5' not found
```

```
## Warning in rm("vst_correlations", "vst_earli_groups", "vst_matrix",
## "select_idx", : object 'p6' not found
```

DESeq analysis with outliers removed

```
# recreate the DESeq object
dds_earli_groups_noOutliers <-
    DESeqDataSetFromMatrix(countData = df_counts_30plus_noOutliers,
                           colData = df_metadata_filtered_g124_noOutliers,
                           design = ~ Group)

# perform the DESeq analysis
dds_earli_groups_noOutliers <- DESeq(dds_earli_groups_noOutliers)
```

```
## estimating size factors
```

```
## estimating dispersions
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
```

```
## final dispersion estimates
```

```
## fitting model and testing
```

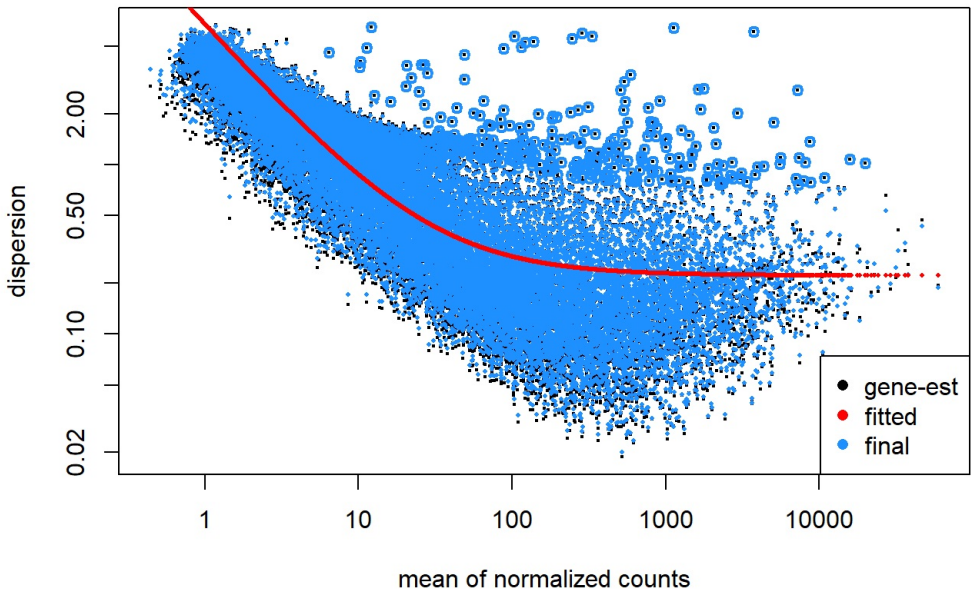
```
## -- replacing outliers and refitting for 194 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)
```

```
## estimating dispersions
```

```
## fitting model and testing
```

Model diagnostics

```
# plot the dispersion estimates to assess quality of the analysis
plotDispEsts(dds_earli_groups_noOutliers)
```



The dispersion should decrease consistently as the mean of the normalized counts increases. The plot above shows an appropriate plot that means the data pass this diagnostic step and the analysis can continue as planned.

Group 1 (Bacteremic sepsis) vs. group 4 (no sepsis) analysis

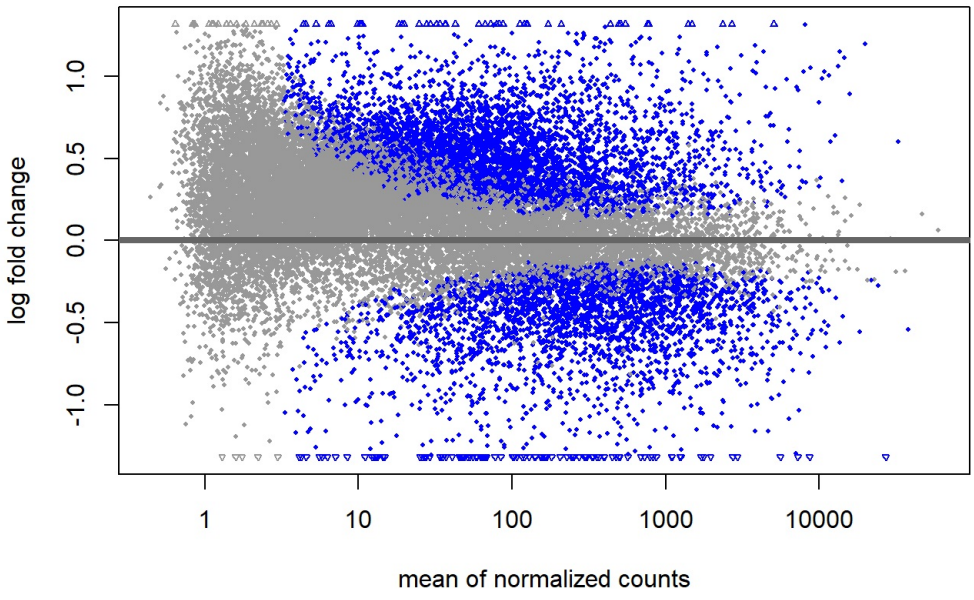
```
# extract group 1 vs. group 4 results
earli_groups14_results <- results(dds_earli_groups_noOutliers,
                                  name = "Group_group_4_vs_group_1",
                                  alpha = 0.05)
```

More model diagnostics

MA-plot (intensity ratio by average intensity)

The `plotMA` function shows the $\log_2$ fold changes attributable to a given variable over the mean of normalized counts for all the samples in `earli_groups14_results`. Points are colored red if the adjusted p value is less than 0.1. Points which fall out of the window are plotted as open triangles pointing either up or down.

```
#* MA plot without LFC shrinkage
plotMA(earli_groups14_results)
```

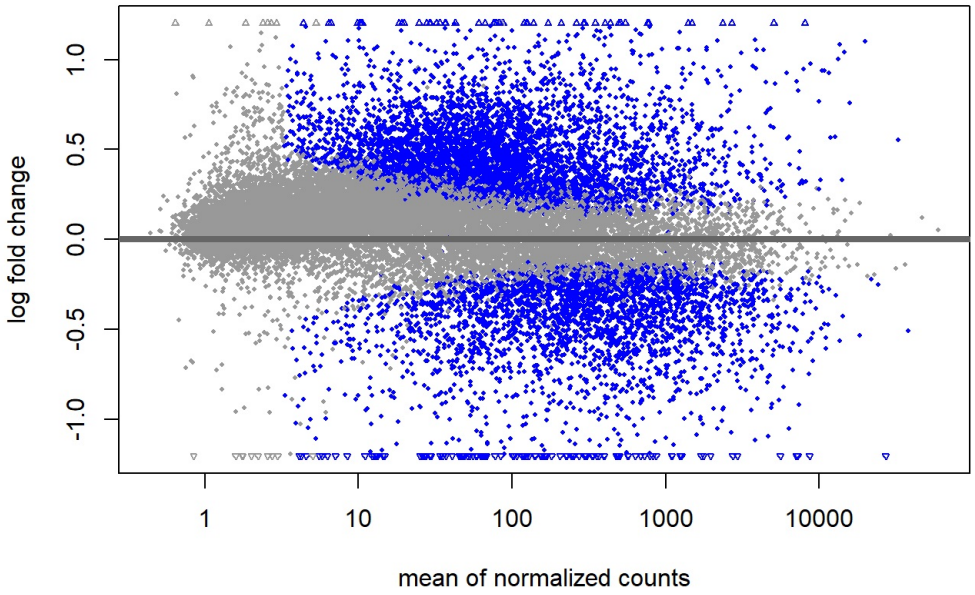


$\log_2$ fold changes of those genes with very low normalized gene counts are shrunken to remove the noise inherent in such low gene count data. The MA-plot is the reproduced to assess these changes.

```
#* perform the LFC shrinkage
earli_groups14_results <- lfcShrink(dds = dds_earli_groups_noOutliers,
                                    coef = "Group_group_4_vs_group_1",
                                    res = earli_groups14_results)
```

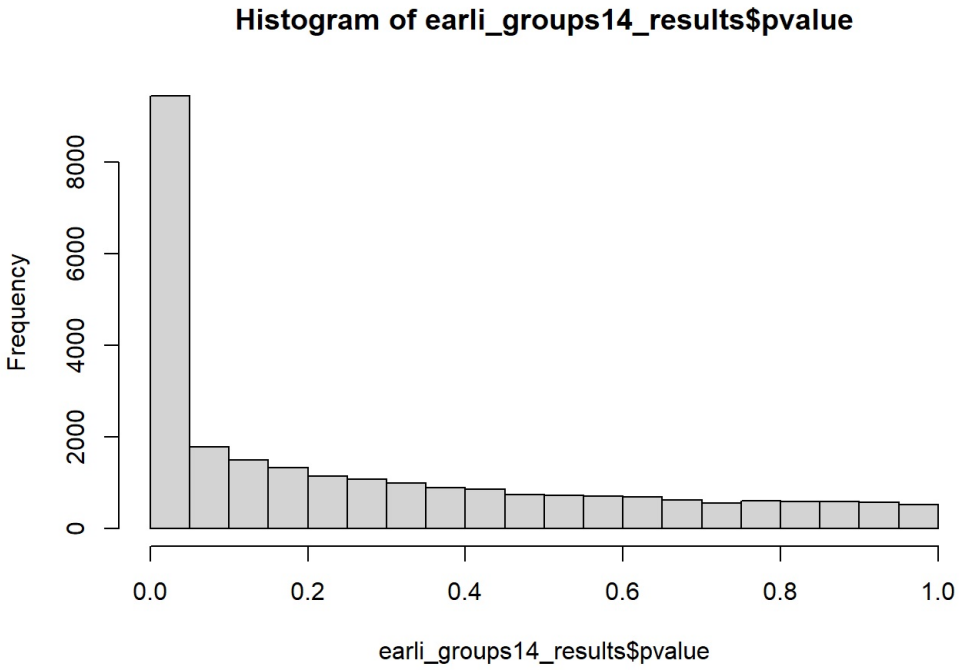
```
## using 'apeglm' for LFC shrinkage. If used in published research, please cite:
##     Zhu, A., Ibrahim, J.G., Love, M.I. (2018) Heavy-tailed prior distributions for
##     sequence count data: removing the noise and preserving large differences.
##     Bioinformatics. https://doi.org/10.1093/bioinformatics/bty895
```

```
#* MA plot after LFC shrinkage
plotMA(earli_groups14_results)
```



p value histogram

```
#* plot a histogram of the p values
hist(earli_groups14_results$pvalue)
```



The large p values should be uniformly distributed while the low p values should increase sharply. This is another model diagnostic that passes in this case.

Summary of results

```
summary(earli_groups14_results)
```

```
##
## out of 25953 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)       : 4491, 17%
## LFC < 0 (down)     : 2995, 12%
## outliers [1]       : 0, 0%
## low counts [2]     : 5032, 19%
## (mean count < 3)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

Generate heatmap figures

Filter for genes with a p value < 0.05

```
# turn the results into a data.table
dt_earli_groups14_results <- data.table(data.frame(earli_groups14_results),
                                        keep.rownames = TRUE)

# change the name of the "rn" row
setnames(dt_earli_groups14_results, old="rn",new="ensg_name")

# order by padj
setorder(dt_earli_groups14_results, padj)

# remove the NA values from the data.table
dt_earli_groups14_results_noNA <- dt_earli_groups14_results[!is.na(padj)]

# save only those adjusted p values < 0.05
dt_earli_groups14_results_noNA_sigP <- dt_earli_groups14_results_noNA[padj < 0.05]
```

Get normalized count values only in the significantly DE genes

```
# filter for genes pval <0.05
normalized_counts_sig <- counts(dds_earli_groups_noOutliers, normalized = TRUE)

# filter all normalized counts using ranked and significant g14 data
normalized_counts_sig <- normalized_counts_sig[dt_earli_groups14_results_noNA_sigP$ensg_name,]

# get normalized_counts for only groups 1 and 4
group14_idx <-
    which(df_metadata_filtered_g124_noOutliers$Group %in% c("group_1", "group_4"))

# filter the data.frame with the index
df_metadata_filtered_g14 <- df_metadata_filtered_g124_noOutliers[group14_idx,]

# drop the empty factor levels
df_metadata_filtered_g14$Group <- droplevels(df_metadata_filtered_g14$Group)

# get the rownames, which can filter the counts matrix
to_keep_14 <- rownames(df_metadata_filtered_g14)

# filter the normalized counts matrix for only groups 1, 4
normalized_counts_sig14 <- normalized_counts_sig[,to_keep_14]
```

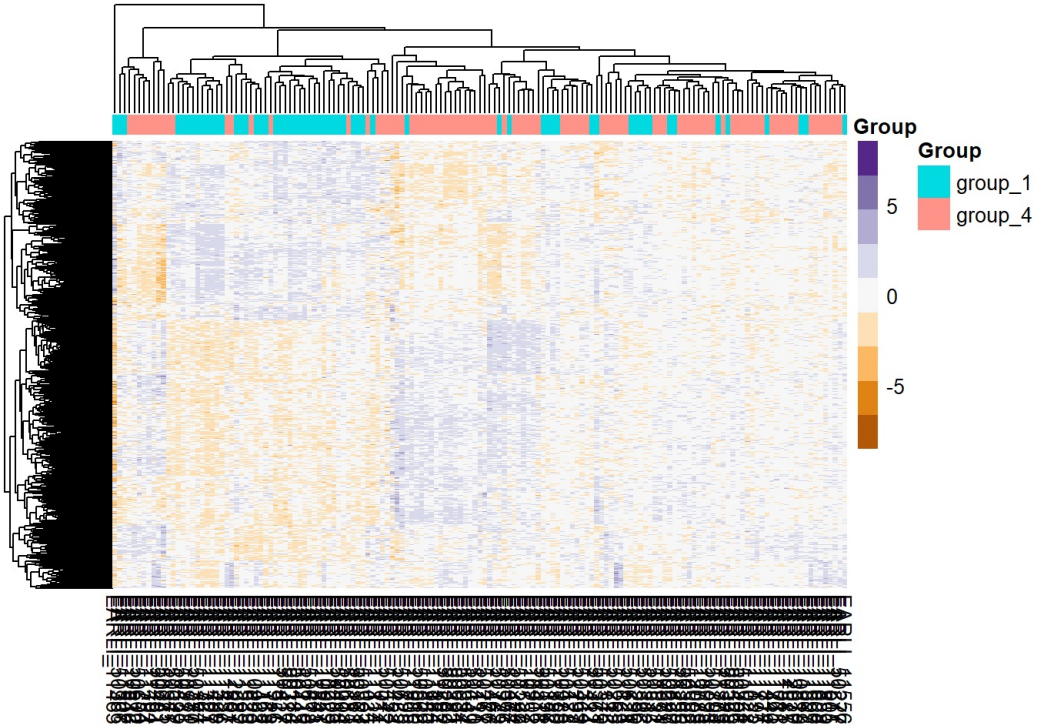
Perform vst transformation

```
# get VST transformed counts
vst_counts <- varianceStabilizingTransformation(dds_earli_groups_noOutliers)
vst_counts_mat <- assay(vst_counts)

# vst_counts_mat has the same exact structure as the normalized_counts_sig matrix above. The same process will be
used to generate the vst matrix to create the heat maps below
vst_counts_mat_sig <- vst_counts_mat[dt_earli_groups14_results_noNA_sigP$ensg_name,]
vst_counts_mat_sig14 <- vst_counts_mat_sig[,to_keep_14]
```

Generate the heatmaps

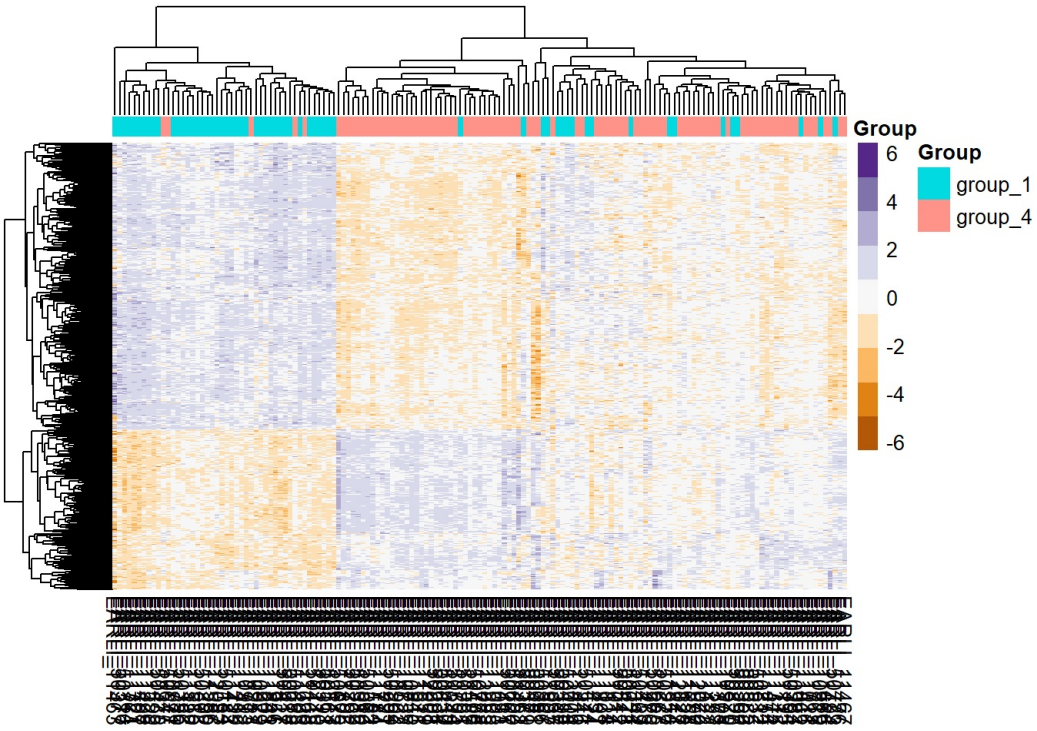
All differentially expressed genes

```
# brew the colors
heat_colors <- brewer.pal(9, "PuOr")

# all vst counts for groups 1 and 4; no top DE genes filtered
pheatmap(vst_counts_mat_sig14,
         color=heat_colors,
         show_rownames = FALSE,
         cluster_rows=TRUE,
         annotation_col=df_metadata_filtered_g14[, c("Group"), drop=FALSE],
         scale="row")
```



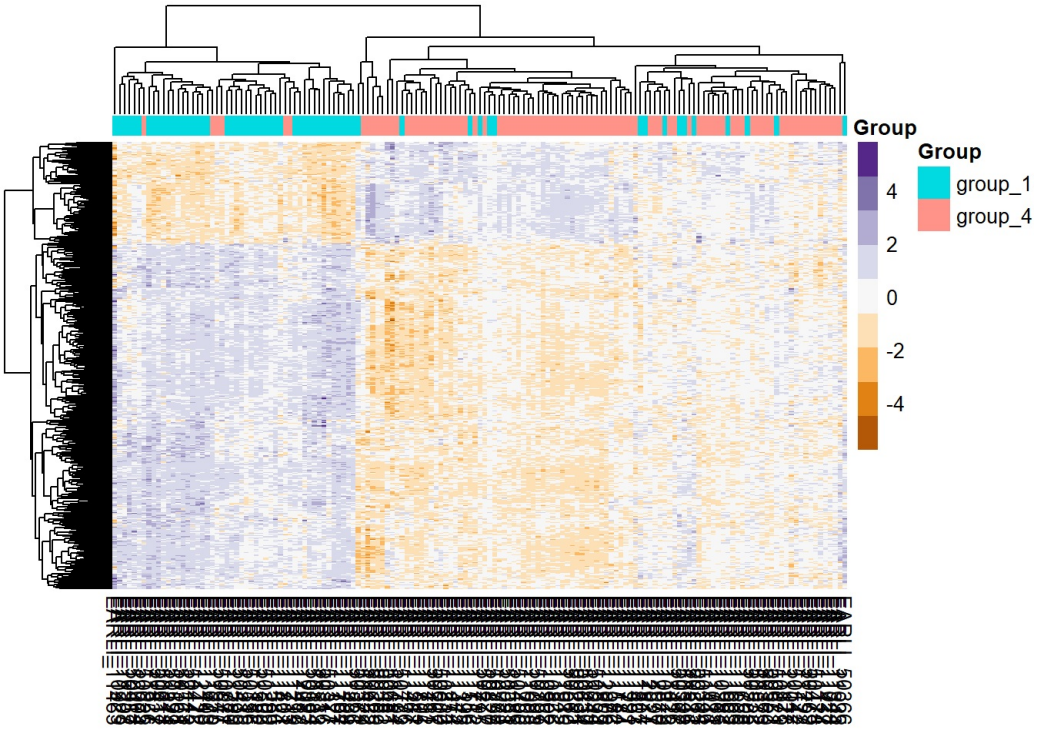
Top 1000 differentially expressed genes

```
# get the top 1000 DE genes from the results
dt_earli_groups14_results_noNA_sigP_top1000 <-
    dt_earli_groups14_results_noNA_sigP[1:1000,]

# get top 1000 for vst normalized counts
vst_counts_mat_sig14_top1000 <-
    vst_counts_mat_sig14[dt_earli_groups14_results_noNA_sigP_top1000$ensg_name,]

pheatmap(vst_counts_mat_sig14_top1000,
         color=heat_colors,
         show_rownames=FALSE,
         cluster_rows=TRUE,
         annotation_col=df_metadata_filtered_g14[, c("Group"), drop=FALSE],
         scale="row")
```



Top 500 differentially expressed genes

```
# get the top 500 DE genes from the results
dt_earli_groups14_results_noNA_sigP_top500 <-
    dt_earli_groups14_results_noNA_sigP[1:500,]

# get top 500 for vst normalized counts
vst_counts_mat_sig14_top500 <-
    vst_counts_mat_sig14[dt_earli_groups14_results_noNA_sigP_top500$ensg_name,]

pheatmap(vst_counts_mat_sig14_top500,
         color=heat_colors,
         show_rownames=FALSE,
         cluster_rows=TRUE,
         annotation_col=df_metadata_filtered_g14[, c("Group"), drop=FALSE],
         scale="row")
```

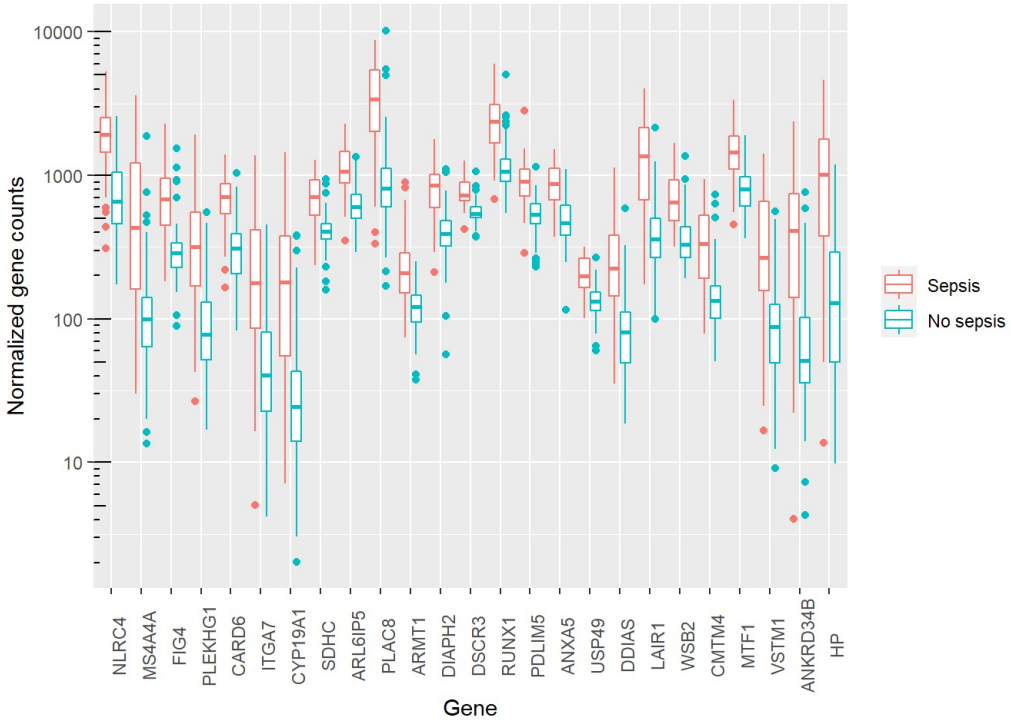


Top 25 differentially expressed genes, boxplot

```
# read in the gene symbols that assess function
gene_symbols <- fread("gene_attr.txt", header = FALSE)
setnames(gene_symbols,
         old=c("V1","V2","V3","V4"),
         new=c("ensg_name","V2","type","symbol"))

# get the top 25 genes based on padj
dt_earli_groups14_results_noNA_sigP_top25 <- dt_earli_groups14_results_noNA_sigP[1:25,]

# subset the normalized counts matrix using the ensg_name
normalized_counts_sig_top25 <-
    normalized_counts_sig[dt_earli_groups14_results_noNA_sigP_top25$ensg_name,]

# make it a data.table
dt_normalized_counts_sig_top25 <- data.table(normalized_counts_sig_top25,
                                             keep.rownames = TRUE)
# rename
setnames(dt_normalized_counts_sig_top25, old="rn", new="ensg_name")

# melt
dt_norm_cnt_sig_top25_melted <- melt(dt_normalized_counts_sig_top25,
                                     id.vars = "ensg_name",
                                     variable.name = "participant",
                                     value.name = "norm_counts")

# merge with the metadata
#* change the metadata into a data.table with only 2 columns
dt_metadata_forMerge <- data.table(df_metadata_filtered_g124_noOutliers,
                                   keep.rownames = TRUE)
setnames(dt_metadata_forMerge, old="rn", new = "participant")
dt_metadata_forMerge <- dt_metadata_forMerge[,c("participant", "Group")]

dt_for_plot <- merge(dt_norm_cnt_sig_top25_melted, dt_metadata_forMerge,
      by = "participant", all.x = TRUE)

dt_for_plot <- merge(dt_for_plot, gene_symbols[,c("ensg_name", "symbol")],
                     by="ensg_name", all.x=TRUE)

# make the plot
ggplot(dt_for_plot[Group %in% c("group_1","group_4")],
       aes(x=symbol,y=norm_counts,
           color=factor(Group,levels=c("group_1","group_4"),
                        labels=c("Sepsis","No sepsis"))))+ # relabel the legend
    geom_boxplot()+
    scale_y_log10()+
    annotation_logticks(sides = "l")+ # add logticks so log scale is obvious
    xlab("Gene")+
    ylab("Normalized gene counts")+
    theme(axis.text.x=element_text(angle=90))+ # angle the axis labels
    scale_x_discrete(limits=c(unique(dt_for_plot$symbol)))+ # order top 1-25
    guides(color=guide_legend(title=NULL)) # remove the legend title
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 4 rows containing non-finite values (stat_boxplot).
```



Webgestalt data preparation

```
#* copy the results without NAs
web_gestalt_g14 <- copy(dt_earli_groups14_results_noNA)

# rank in order of logFC from highest to lowest
web_gestalt_g14_p0.1 <- web_gestalt_g14[padj <0.1]

# plug in two columns; first column is gene name, 2nd column in logFC
web_gestalt_g14_p0.1 <- merge(web_gestalt_g14_p0.1, gene_symbols, by="ensg_name", all.x=T)
web_gestalt_g14_p0.1[,`:=`(baseMean=NULL, lfcSE=NULL, pvalue=NULL, padj=NULL,
                      V2=NULL,ensg_name=NULL,type=NULL)]
setcolorder(web_gestalt_g14_p0.1,neworder = c("symbol", "log2FoldChange"))
setorder(web_gestalt_g14_p0.1, log2FoldChange)
write.table(web_gestalt_g14_p0.1, "webgestalt_g14.rnk", sep = "\t", row.names = F,
            col.names=F, quote=F)

# write the table to plug into webgestalt
write.table(web_gestalt_g14_p0.1,
            "webgestalt_g14_foldSwitch.rnk", sep = "\t", row.names = F,
            col.names=F, quote=F)
```

Table 1 code

```
library(data.table)
library(tableone)

# read in the data ####
data_forTable1 <- fread("EARLI data merged_all clinical_all past medical_Chaz composite included_virusesPresent_M
TA_9-29-2020.csv")

# remove V1 column, meaningless
data_forTable1[,V1:=NULL]

# look at the column names to see which to keep
colnames(data_forTable1)
```

```
##   [1] "EARLIStudyId"                  "PatientID"
##   [3] "Age"                           "Gender"
##   [5] "Race"                          "Caucasian_Ethnicity"
##   [7] "Asian_Ethnicity"               "HospWithin30d"
##   [9] "PrimaryDiag"                   "PNA_community"
##  [11] "PNA_healthcare"                "Aspiration"
##  [13] "Asthma"                        "COPD2"
##  [15] "PulmonaryEmbolism"             "DeepVenousThrombosis"
##  [17] "InterstitialLungDisease"       "Pneumothorax"
##  [19] "LungTransplant"                "CysticFibrosis"
##  [21] "NeuromuscularWeakness"         "HypoxicRespFailure"
```

```
##  [23] "HypercardicRespFailure"          "ChronicVentilatorDependence"
##  [25] "ObstructiveSleepApnea"           "ShortnessOfBreath"
##  [27] "Hemoptysis"                      "AcuteCoronarySyndrome"
##  [29] "CoronaryArteryDisease"           "CongestiveHeartFailure"
##  [31] "AtrialArrhythmia"                "VentricularArrhythmia"
##  [33] "ChestPain"                       "PericardialEffusion"
##  [35] "Syncope"                         "PulmonaryHypertension"
##  [37] "HeartTransplant"                 "CardiacSurgery"
##  [39] "HypertensiveCrisis"              "PeripheralVascularDisease"
##  [41] "AAA"                             "AorticDissection"
##  [43] "Hypertension2"                   "Hypotension"
##  [45] "Tachycardia"                     "Bradycardia"
##  [47] "ValvularDisease"                 "PulmonaryEdema"
##  [49] "CardiacArrest"                   "Upper_GI_bleed"
##  [51] "Lower_GI_bleed"                  "EndStageLiverDisease"
##  [53] "FulminantHepaticFailure"         "HepatitisC"
##  [55] "HepatitisB"                      "AlcoholicHepatistis"
##  [57] "Cirrhosis2"                      "AlcoholicCirrhosis2"
##  [59] "HepatocellularCarcinoma"         "Cholangitis_cholecystitis"
##  [61] "LiverTransplant"                 "PepticUlcerDisease"
##  [63] "BowelIschemia_infarction"        "Obstruction"
##  [65] "PtPerforatedViscus"              "Ileus"
##  [67] "Pancreatitis"                    "Diarrhea"
##  [69] "ShockLive"                       "GERD"
##  [71] "EndStageRenalDz"                 "ChronicKidneyDisease"
##  [73] "AcuteRenalFailure"               "MetabolicAcidosis"
##  [75] "MetabolicAlkalosis"              "KidneyTransplant"
##  [77] "VolumeOverload"                  "UrinaryObstruction"
##  [79] "Uremia"                          "Rhabdomyolysis"
##  [81] "Anemia"                          "Thrombocytopenia"
##  [83] "PtSepsis"                        "Influenza"
##  [85] "Meningitis"                      "Endocarditis"
##  [87] "LineInfection"                   "Skin_softTissue_infection"
##  [89] "HIV_AIDS"                        "CD4"
##  [91] "OtherChronicImmunosuppresion"    "BacterialNOS"
##  [93] "ViralNOS"                        "Fever"
##  [95] "UTI"                             "c_Diff_colitis"
##  [97] "IntraabdominalInfection"         "FebrileNeutropenia"
##  [99] "AcuteLeukemia"                   "ChronicLeukemia"
## [101] "BlastCrisis"                     "BoneMarrow_StemCellTransplant"
## [103] "SolidTumor"                      "TumorOrgan"
## [105] "Metastatic"                      "LungCancer"
## [107] "TTP_HUS"                         "ITP"
## [109] "OtherHematologicMalignancy"      "Hyperkalemia"
## [111] "Hypokalemia"                     "Hypernatremia"
## [113] "Hyponatremia"                    "Hypercalcemia"
## [115] "Hypocalcemia"                    "Hypermagnesemia"
## [117] "Hypomagnesemia"                  "Hyperthyroid"
## [119] "Hypothyroid"                     "AdrenalInsufficiency"
## [121] "Hyperglycemia"                   "Hypoglycemia"
## [123] "DM_type_I"                       "DM_type_II"
## [125] "DKA"                             "HONK"
## [127] "IntraparenchymalHemorrhage"      "SubarachnoidHemorrhage"
## [129] "SubduralHemorrhage"              "CVA"
## [131] "AlteredMentalStatus"             "ArteriovenousMalformation"
## [133] "Delirium"                        "CNS_tumor"
## [135] "Weakness"                        "Seizure"
## [137] "Dementia"                        "HepaticEncephalopathy"
## [139] "SLE"                             "RheumatoidArthritis"
## [141] "OtherRheumatologic"              "Schizophrenia"
## [143] "BipolarDisorder"                 "AnxietyDisorder"
## [145] "Depression"                      "DrugOverdose"
## [147] "Pregnant"                        "PostPartum"
## [149] "AlcoholIntoxication"             "AlcoholWithdrawal"
## [151] "UpperAirwayObstruction"          "AlcoholAbuse2"
## [153] "Other_substance_abuse"           "OtherDisease"
## [155] "Group"                           "patientid_main"
## [157] "Barcode"                         "APACHEIII"
## [159] "28d death"                       "60d death"
## [161] "Hospital death"                  "MechVent"
## [163] "Blood"                           "Lungs"
## [165] "Urine"                           "Stool"
## [167] "Other"                           "TempMaxSAPS"
## [169] "TempMinSAPS"                     "WBCMaxSAPS"
## [171] "WBCMinSAPS"                      "HRMaxSAPS"
## [173] "HRMinSAPS"                       "RRMaxSAPS"
## [175] "RRMinSAPS"                       "SIRS_HR"
## [177] "SIRS_temp"                       "SIRS_RR"
## [179] "SIRS_WBC"                        "SIRS_total"
```

```
## [181] "SBPMinSAPS"                       "SBPMaxSAPS"
## [183] "CreatinineMaxSAPS"                "CreatinineMinSAPS"
## [185] "PlateletsMinSAPS"                 "APACHEII"
## [187] "Intubated"                        "OnPressorsSAPS"
## [189] "ER_admit_date"                    "ER_discharge_date"
## [191] "ICU_admit_date"                   "BirthDate"
## [193] "HOST_PAXgene_filename"            "MICROBE_Plasma_DNA-Seq_filename"
## [195] "MICROBE_Plasma_RNA-Seq_filename" "ER_admit_diag"
## [197] "ICU_admit_diag"                   "virusPresent"
```

```
# index of columns to keep for table 1
index <- c(1,3,4,5,6,7,8,10,11,14,19,30,37,39,43,49,52,53,57,58,61,71,72,76,83,
           89,90,91,98,99,100,101,102,103,104,105,106,109,123,124,125,126,134,
           138,139,140,141,155,158,159,160,161,162,168:189,192,198)

# these are the column names to keep for table 1 ####
colnames(data_forTable1)[index]
```

```
##  [1] "EARLIStudyId"                   "Age"
##  [3] "Gender"                         "Race"
##  [5] "Caucasian_Ethnicity"            "Asian_Ethnicity"
##  [7] "HospWithin30d"                  "PNA_community"
##  [9] "PNA_healthcare"                 "COPD2"
## [11] "LungTransplant"                 "CongestiveHeartFailure"
## [13] "HeartTransplant"                "HypertensiveCrisis"
## [15] "Hypertension2"                  "CardiacArrest"
## [17] "EndStageLiverDisease"           "FulminantHepaticFailure"
## [19] "Cirrhosis2"                     "AlcoholicCirrhosis2"
## [21] "LiverTransplant"                "EndStageRenalDz"
## [23] "ChronicKidneyDisease"           "KidneyTransplant"
## [25] "PtSepsis"                       "HIV_AIDS"
## [27] "CD4"                            "OtherChronicImmunosuppresion"
## [29] "FebrileNeutropenia"             "AcuteLeukemia"
## [31] "ChronicLeukemia"                "BlastCrisis"
## [33] "BoneMarrow_StemCellTransplant" "SolidTumor"
## [35] "TumorOrgan"                     "Metastatic"
## [37] "LungCancer"                     "OtherHematologicMalignancy"
## [39] "DM_type_I"                      "DM_type_II"
## [41] "DKA"                            "HONK"
## [43] "CNS_tumor"                      "HepaticEncephalopathy"
## [45] "SLE"                            "RheumatoidArthritis"
## [47] "OtherRheumatologic"             "Group"
## [49] "APACHEIII"                      "28d death"
## [51] "60d death"                      "Hospital death"
## [53] "MechVent"                       "TempMaxSAPS"
## [55] "TempMinSAPS"                    "WBCMaxSAPS"
## [57] "WBCMinSAPS"                     "HRMaxSAPS"
## [59] "HRMinSAPS"                      "RRMaxSAPS"
## [61] "RRMinSAPS"                      "SIRS_HR"
## [63] "SIRS_temp"                      "SIRS_RR"
## [65] "SIRS_WBC"                       "SIRS_total"
## [67] "SBPMinSAPS"                     "SBPMaxSAPS"
## [69] "CreatinineMaxSAPS"              "CreatinineMinSAPS"
## [71] "PlateletsMinSAPS"               "APACHEII"
## [73] "Intubated"                      "OnPressorsSAPS"
## [75] "ER_admit_date"                  "BirthDate"
## [77] "virusPresent"
```

```
data_forTable_1 <- data_forTable1[, index, with=FALSE]

# check that the column names wanted were kept
identical(colnames(data_forTable1)[index], colnames(data_forTable_1))
```

```
## [1] TRUE
```

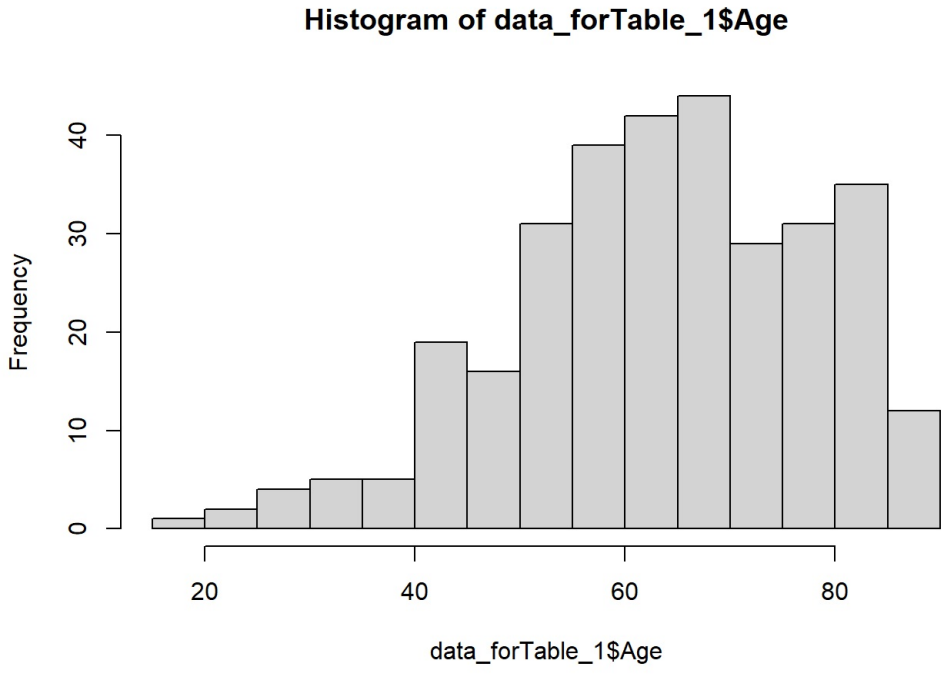
```
# make the date columns into Date objects ####
data_forTable_1$ER_admit_date <- as.Date(data_forTable_1$ER_admit_date,
                                        format = "%m/%d/%Y")

data_forTable_1$BirthDate <- as.Date(data_forTable_1$BirthDate,
                                     format = "%m/%d/%Y")

# how many if each variable of interest is missing ####
# initialize vector to place missing totals into
missing_totals <- vector(mode="numeric")
for(x in 1:ncol(data_forTable_1)){
    i<-sum(is.na(data_forTable_1[,x,with=FALSE]))
    missing_totals[x] <- i
}

# add the first column back to allow keeping the EARLI IDs in the following subset
# add the ER_admit_date column (#75) to keep this column for age calculations
missing_totals[c(1,75)] <- 1

# index of columns with missing values + EARLI ID column from manipulation above
missing_cols <- which(missing_totals>0)

# number of missing in each of those columns
missing_totals[missing_cols]
```

```
##  [1]   1  11   5 326 309 277   2   1   1   1   2
```

```
# subset of data.table with missing values ####
data_forTable_1_missing <- data_forTable_1[,missing_cols,with=FALSE]

# remove the ethnicity columns, these are redundant given race is available ####
data_forTable_1_missing <- data_forTable_1_missing[,-c(3:4)]
data_forTable_1 <- data_forTable_1[,`:=`(Caucasian_Ethnicity=NULL,
                                         Asian_Ethnicity=NULL)]

# calculate missing ages ####
data_forTable_1_missing[, age_calc := ER_admit_date - BirthDate]
data_forTable_1_missing[, age_calc := trunc(as.numeric(age_calc/365.25))]

# no more NAs
sum(is.na(data_forTable_1$Age))
```

```
## [1] 11
```

```
# there are some ages that are 0-10
hist(data_forTable_1$Age)
```



```
# one person whose Age is 0 and the ER_admit_date and BirthDate don't make sense ####
data_forTable_1[Age<18] # EARLI ID 615
```

```
## Empty data.table (0 rows and 75 cols): EARLIStudyId,Age,Gender,Race,HospWithin30d,PNA_community...
```

```
# looked in Quesgen, EARLI ID 615 has age 90

# two patients of unclear mech vent status, EARLI IDs 5366 5492
data_forTable_1_missing[which(is.na(MechVent))]$EARLIStudyId
```

```
## [1] 5366 5492
```

```
# looked in Quesgen, EARLI ID 5366 was mechanically ventilated
# looked in Quesgen, EARLI ID 5492 was mechanically ventilated

# one patient with unclear TempMaxSAPS and TempMinSAPS, EARLI ID 810
data_forTable_1_missing[which(is.na(TempMaxSAPS))]$EARLIStudyId
```

```
## [1] 810
```

```
data_forTable_1_missing[which(is.na(TempMinSAPS))]$EARLIStudyId
```

```
## [1] 810
```

```
# looked in Quesgen and Epic, no temperature was recorded during his hospital stay

# fix all the missing values that are able to be fixed ####

#* AGE ####
# replace the Age NAs with age_calc
data_forTable_1$Age <- ifelse(is.na(data_forTable_1$Age),
                              data_forTable_1_missing$age_calc,
                              data_forTable_1$Age)

# looked in Quesgen, EARLI ID 615 has age 90
data_forTable_1[Age < 18]$Age <- 90
hist(data_forTable_1$Age)

#* MECHVENT ####
# looked in Quesgen, EARLI ID 5366 was mechanically ventilated
# looked in Quesgen, EARLI ID 5492 was mechanically ventilated
data_forTable_1[is.na(MechVent)]$MechVent <- 1
data_forTable_1[EARLIStudyId==5366]$MechVent
```

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

```
data_forTable_1[EARLIStudyId==5492]$MechVent
```

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

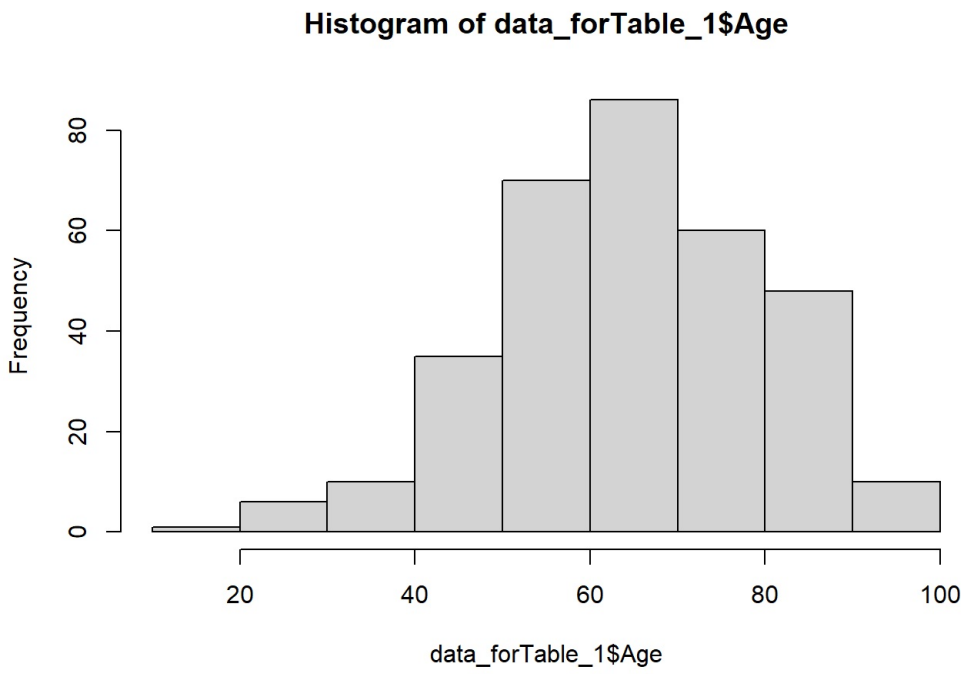
```
#* CD4 count ####
# in order to allow for a simple calculation of immunocompromised HIV pts CD4 <200
# will replace all NAs with 9999
data_forTable_1[is.na(CD4)]$CD4 <- 9999
data_forTable_1[CD4==">1000"]$CD4 <- 1001
data_forTable_1[CD4==">500"]$CD4 <- 501
data_forTable_1$CD4  <- as.numeric(data_forTable_1$CD4)

# combine all the redundant data into columns for Table 1 ####
colnames(data_forTable_1)
```

```
##  [1] "EARLIStudyId"                  "Age"
##  [3] "Gender"                        "Race"
##  [5] "HospWithin30d"                 "PNA_community"
##  [7] "PNA_healthcare"                "COPD2"
##  [9] "LungTransplant"                "CongestiveHeartFailure"
## [11] "HeartTransplant"               "HypertensiveCrisis"
## [13] "Hypertension2"                 "CardiacArrest"
## [15] "EndStageLiverDisease"          "FulminantHepaticFailure"
## [17] "Cirrhosis2"                    "AlcoholicCirrhosis2"
## [19] "LiverTransplant"               "EndStageRenalDz"
## [21] "ChronicKidneyDisease"          "KidneyTransplant"
## [23] "PtSepsis"                      "HIV_AIDS"
## [25] "CD4"                           "OtherChronicImmunosuppresion"
## [27] "FebrileNeutropenia"            "AcuteLeukemia"
## [29] "ChronicLeukemia"               "BlastCrisis"
## [31] "BoneMarrow_StemCellTransplant" "SolidTumor"
## [33] "TumorOrgan"                    "Metastatic"
## [35] "LungCancer"                    "OtherHematologicMalignancy"
## [37] "DM_type_I"                     "DM_type_II"
## [39] "DKA"                           "HONK"
## [41] "CNS_tumor"                     "HepaticEncephalopathy"
## [43] "SLE"                           "RheumatoidArthritis"
## [45] "OtherRheumatologic"            "Group"
## [47] "APACHEIII"                     "28d death"
## [49] "60d death"                     "Hospital death"
## [51] "MechVent"                      "TempMaxSAPS"
## [53] "TempMinSAPS"                   "WBCMaxSAPS"
## [55] "WBCMinSAPS"                    "HRMaxSAPS"
## [57] "HRMinSAPS"                     "RRMaxSAPS"
## [59] "RRMinSAPS"                     "SIRS_HR"
## [61] "SIRS_temp"                     "SIRS_RR"
## [63] "SIRS_WBC"                      "SIRS_total"
## [65] "SBPMinSAPS"                    "SBPMaxSAPS"
## [67] "CreatinineMaxSAPS"             "CreatinineMinSAPS"
## [69] "PlateletsMinSAPS"              "APACHEII"
## [71] "Intubated"                     "OnPressorsSAPS"
## [73] "ER_admit_date"                 "BirthDate"
## [75] "virusPresent"
```

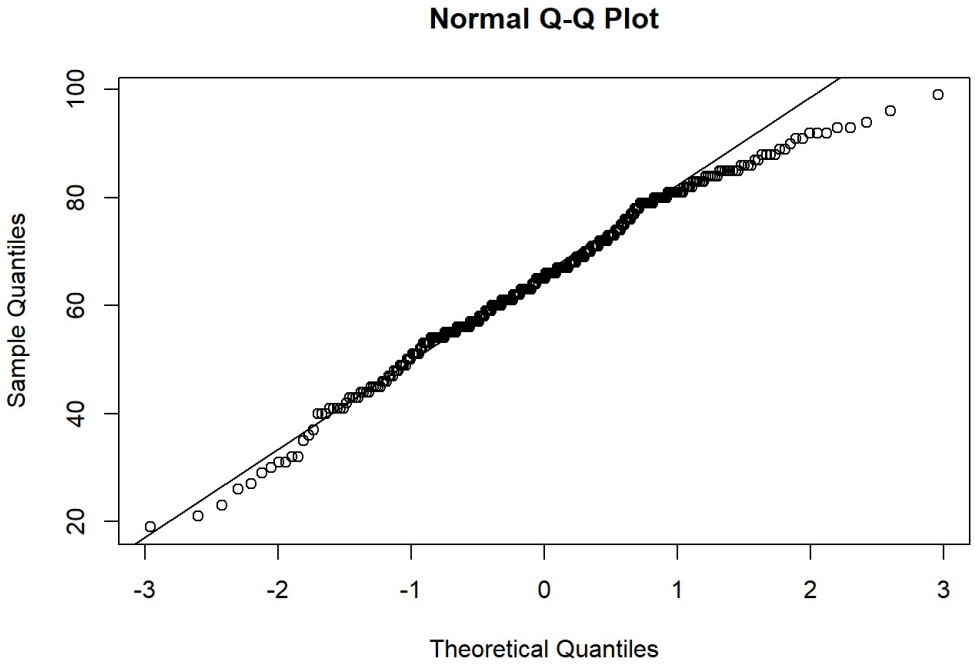
```
# solid organ transplant
data_forTable_1$SOT <- ifelse((data_forTable_1$HeartTransplant==1 |
                                 data_forTable_1$LiverTransplant==1 |
                                 data_forTable_1$LungTransplant==1 |
                                 data_forTable_1$KidneyTransplant==1), 1, 0)

# Hypertension
data_forTable_1$HTN <- ifelse((data_forTable_1$HypertensiveCrisis==1 |
                                 data_forTable_1$Hypertension2==1), 1, 0)

# Cirrhosis
data_forTable_1$Cirrhosis <- ifelse((data_forTable_1$EndStageLiverDisease==1 |
                                       data_forTable_1$FulminantHepaticFailure==1 |
                                       data_forTable_1$Cirrhosis2==1 |
                                       data_forTable_1$AlcoholicCirrhosis2==1 |
                                       data_forTable_1$HepaticEncephalopathy==1), 1, 0)

# End stage renal disease
data_forTable_1$CKD <- ifelse((data_forTable_1$EndStageRenalDz==1 |
                                 data_forTable_1$ChronicKidneyDisease==1), 1, 0)

# Malignancy
data_forTable_1$Malignancy <- ifelse((data_forTable_1$SolidTumor==1 |
                                        !is.na(data_forTable_1$TumorOrgan) |
                                        data_forTable_1$Metastatic==1 |
                                        data_forTable_1$FebrileNeutropenia==1 |
                                        data_forTable_1$BlastCrisis==1 |
                                        data_forTable_1$ChronicLeukemia==1 |
                                        data_forTable_1$OtherHematologicMalignancy==1 |
                                        data_forTable_1$CNS_tumor==1 |
                                        data_forTable_1$LungCancer==1 |
                                        data_forTable_1$AcuteLeukemia), 1, 0)

# Immunocompromised
data_forTable_1$Immunocompromised <- ifelse((data_forTable_1$SOT==1 |
                                               data_forTable_1$SLE==1 |
                                               data_forTable_1$RheumatoidArthritis==1 |
                                               data_forTable_1$OtherRheumatologic==1 |
                                               data_forTable_1$BoneMarrow_StemCellTransplant==1 |
                                               data_forTable_1$FebrileNeutropenia==1 |
                                               data_forTable_1$OtherChronicImmunosuppresion==1 |
                                               data_forTable_1$AcuteLeukemia==1 |
                                               data_forTable_1$ChronicLeukemia==1 |
                                               data_forTable_1$BlastCrisis==1 |
                                               data_forTable_1$OtherHematologicMalignancy==1 |
                                               data_forTable_1$CD4 < 200), 1, 0)

# HIV
data_forTable_1$HIV <- ifelse((data_forTable_1$HIV_AIDS==1 |
                                 data_forTable_1$CD4!=9999), 1, 0)

# Diabetes
data_forTable_1$Diabetes <- ifelse((data_forTable_1$DM_type_I==1 |
                                      data_forTable_1$DM_type_II==1 |
                                      data_forTable_1$HONK==1 |
                                      data_forTable_1$DKA==1), 1, 0)

# remove redundant columns ####
data_forTable_1[, `:=`(HeartTransplant=NULL,LiverTransplant=NULL,LungTransplant=NULL,
                       KidneyTransplant=NULL,HypertensiveCrisis=NULL,Hypertension2=NULL,
                       EndStageLiverDisease=NULL,FulminantHepaticFailure=NULL,
                       Cirrhosis2=NULL,AlcoholicCirrhosis2=NULL,HepaticEncephalopathy=NULL,
                       EndStageRenalDz=NULL,ChronicKidneyDisease=NULL,SolidTumor=NULL,
                       TumorOrgan=NULL,Metastatic=NULL,FebrileNeutropenia=NULL,
                       BlastCrisis=NULL,ChronicLeukemia=NULL,OtherHematologicMalignancy=NULL,
                       CNS_tumor=NULL,LungCancer=NULL,AcuteLeukemia=NULL,SLE=NULL,
                       RheumatoidArthritis=NULL,OtherRheumatologic=NULL,
                       BoneMarrow_StemCellTransplant=NULL,OtherChronicImmunosuppresion=NULL,
                       CD4=NULL,HIV_AIDS=NULL,DM_type_I=NULL,DM_type_II=NULL,HONK=NULL,DKA=NULL)]

# Make multicategory variables factors ####
colnames(data_forTable_1)
```

```
##  [1] "EARLIStudyId"           "Age"                    "Gender"
##  [4] "Race"                   "HospWithin30d"          "PNA_community"
##  [7] "PNA_healthcare"         "COPD2"                  "CongestiveHeartFailure"
## [10] "CardiacArrest"          "PtSepsis"               "Group"
## [13] "APACHEIII"              "28d death"              "60d death"
## [16] "Hospital death"         "MechVent"               "TempMaxSAPS"
## [19] "TempMinSAPS"            "WBCMaxSAPS"             "WBCMinSAPS"
## [22] "HRMaxSAPS"              "HRMinSAPS"              "RRMaxSAPS"
## [25] "RRMinSAPS"              "SIRS_HR"                "SIRS_temp"
## [28] "SIRS_RR"                "SIRS_WBC"               "SIRS_total"
## [31] "SBPMinSAPS"             "SBPMaxSAPS"             "CreatinineMaxSAPS"
## [34] "CreatinineMinSAPS"      "PlateletsMinSAPS"       "APACHEII"
## [37] "Intubated"              "OnPressorsSAPS"         "ER_admit_date"
## [40] "BirthDate"              "virusPresent"           "SOT"
## [43] "HTN"                    "Cirrhosis"              "CKD"
## [46] "Malignancy"             "Immunocompromised"      "HIV"
## [49] "Diabetes"
```

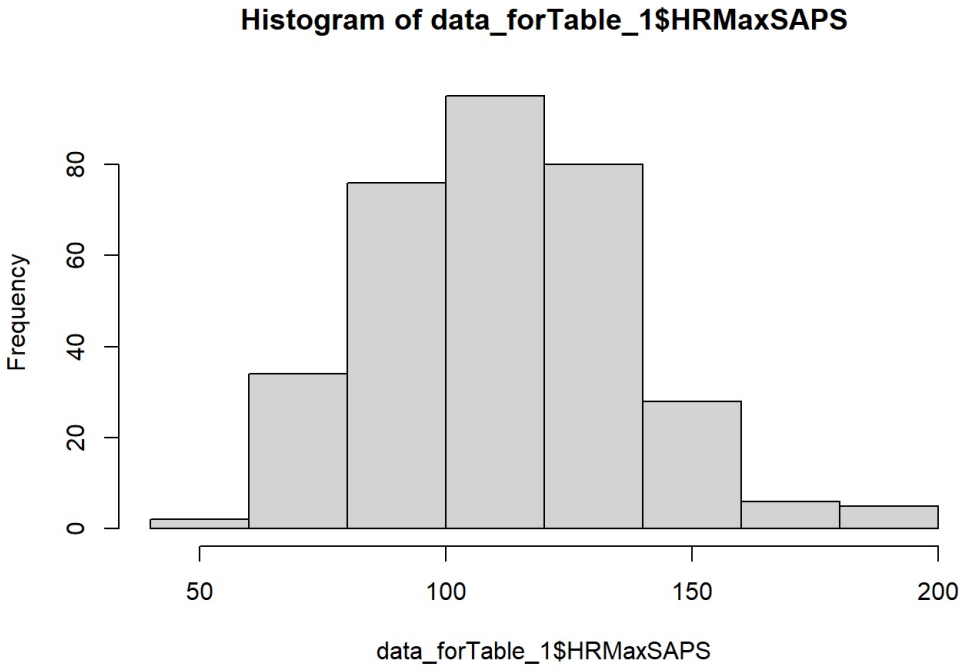
```
# Group
data_forTable_1$Group <- factor(data_forTable_1$Group)

# Gender: 1=male, 2=female, 3=transgender
data_forTable_1$Gender <- factor(data_forTable_1$Gender, levels = c(1,2,3),
                                 labels = c("Male", "Female", "Transgender"))

# Race: 1=Caucasian, 2=African American, 3=Asian, 4=Pacific Islander, 5=Native American
# 6=Other, 7=Unknown
data_forTable_1$Race <- factor(data_forTable_1$Race, levels = c(1,2,3,4,5,6,7),
                               labels = c("Caucasian", "African American", "Asian",
                                          "Pacific Islander", "Native American",
                                          "Other", "Unknown"))

# put together Table 1 ####
# check the distribution of the continuous variables
hist(data_forTable_1$Age)
```



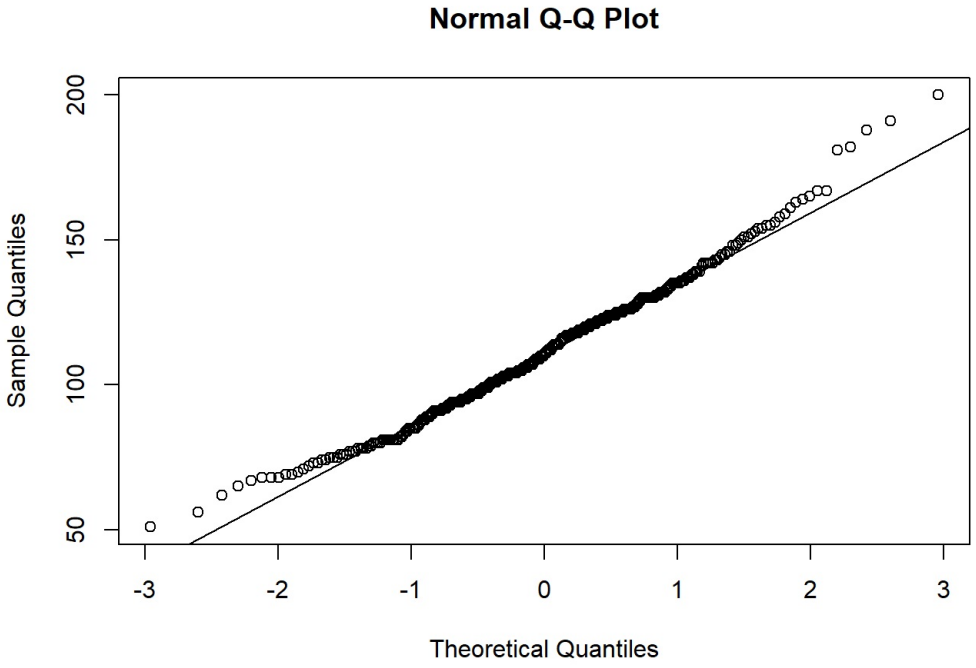

```
qqnorm(data_forTable_1$Age); qqline(data_forTable_1$Age) # normal?
```



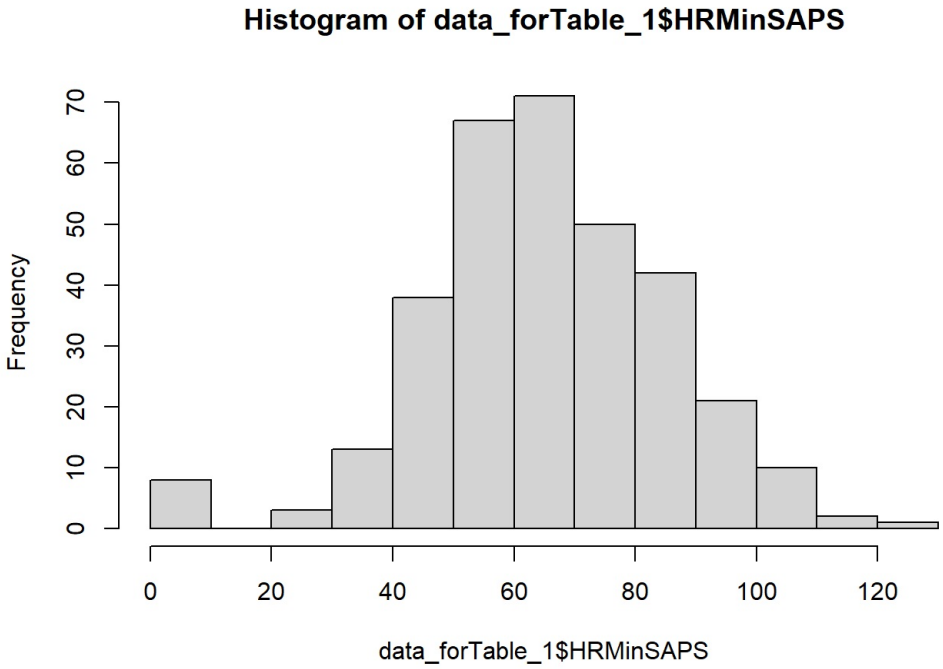

```
hist(data_forTable_1$HRMaxSAPS)
```



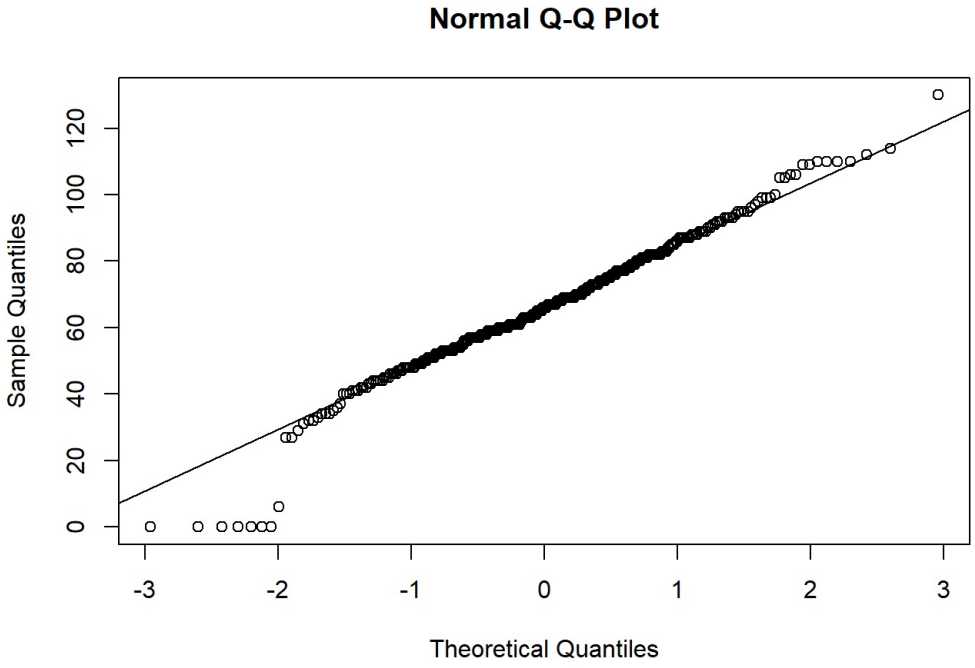

```
qqnorm(data_forTable_1$HRMaxSAPS); qqline(data_forTable_1$HRMaxSAPS) # normal
```



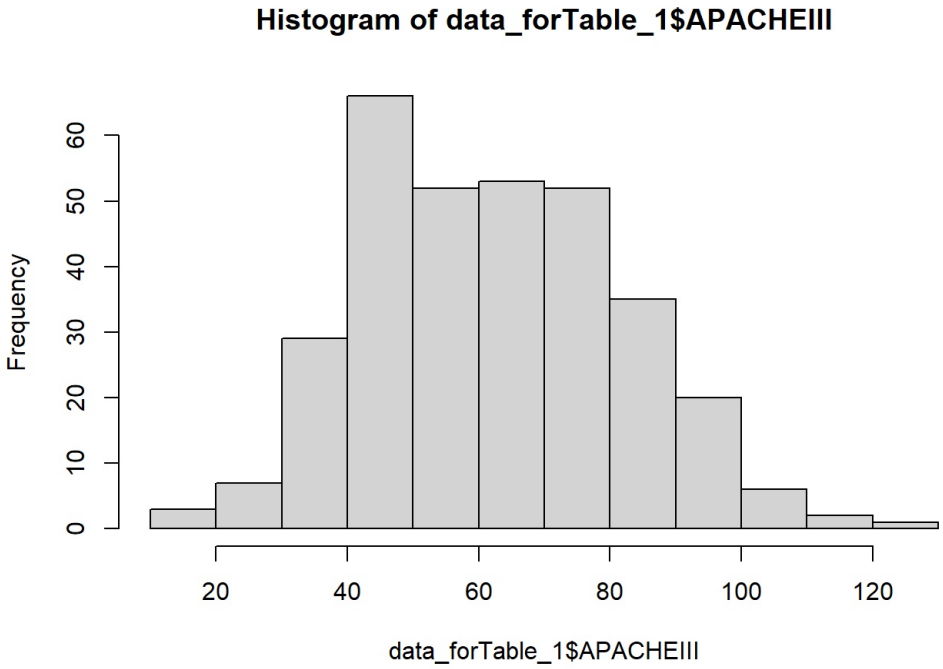

```
hist(data_forTable_1$HRMinSAPS)
```



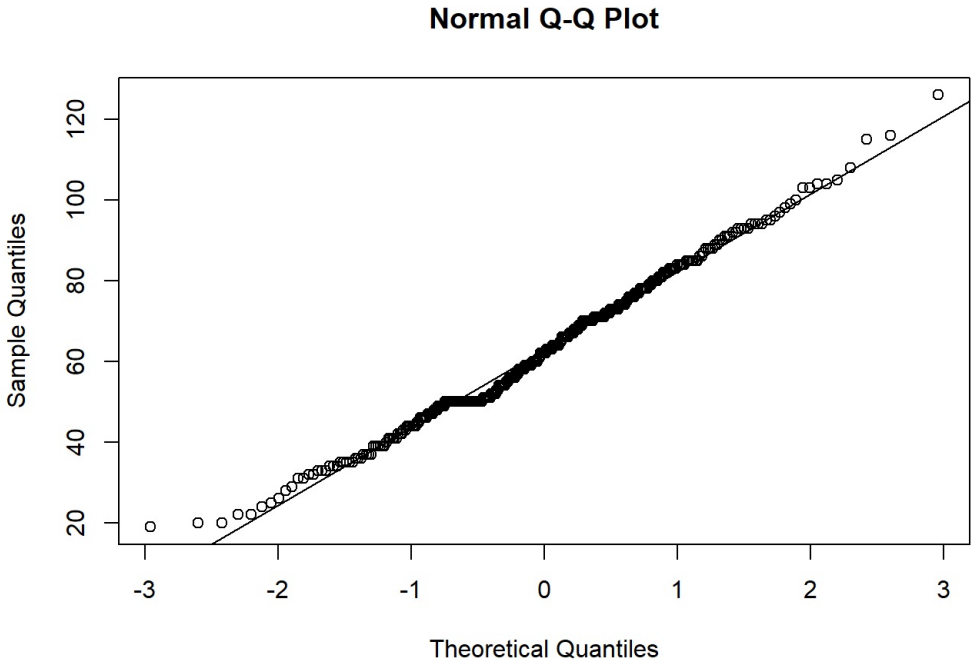

```
qqnorm(data_forTable_1$HRMinSAPS); qqline(data_forTable_1$HRMinSAPS) # normal?
```



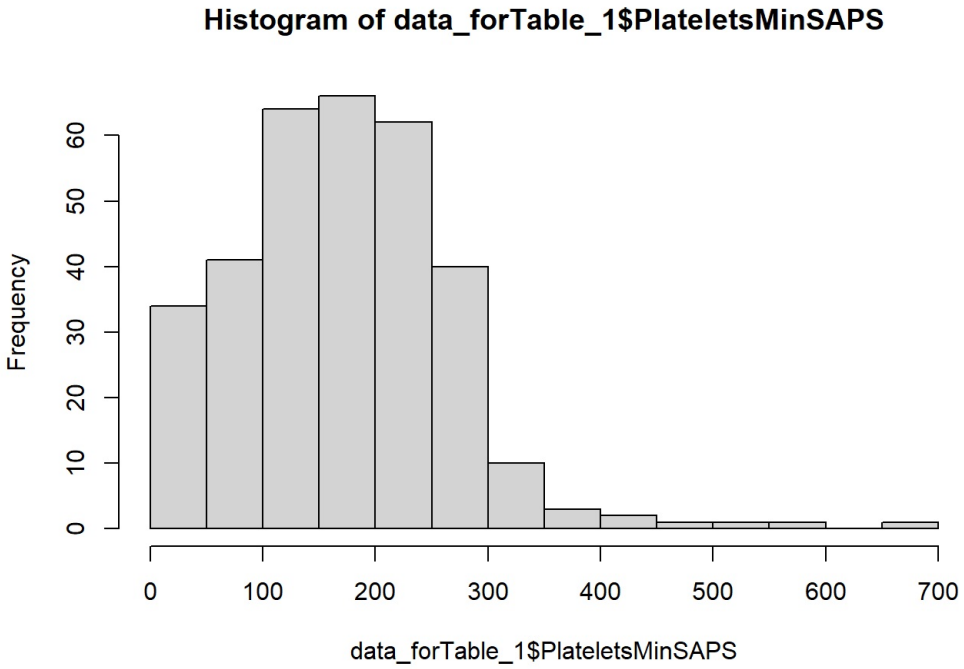

```
hist(data_forTable_1$APACHEIII)
```



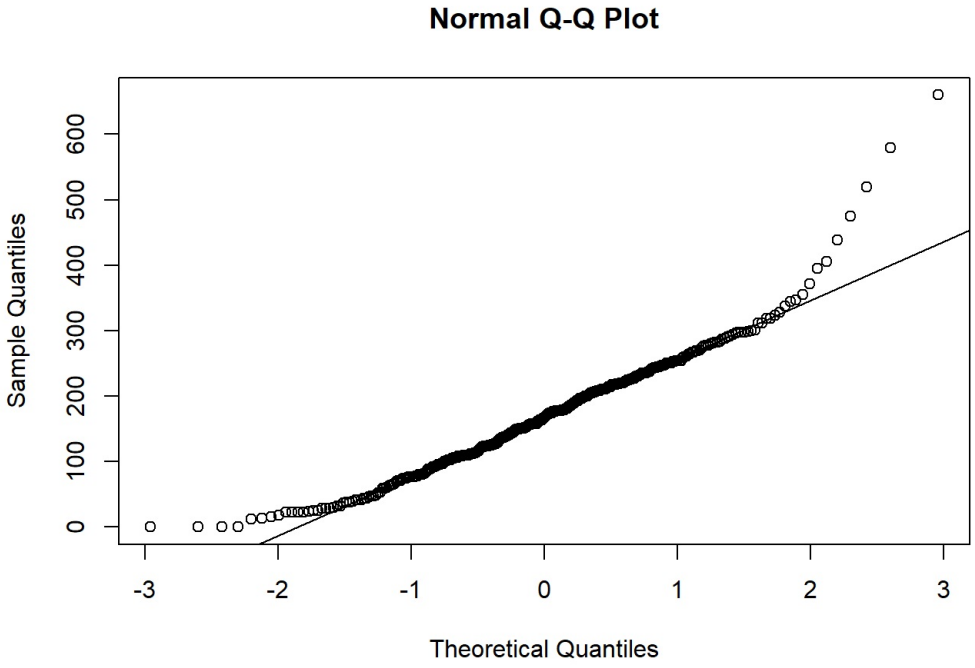

```
qqnorm(data_forTable_1$APACHEIII); qqline(data_forTable_1$APACHEIII) # normal
```



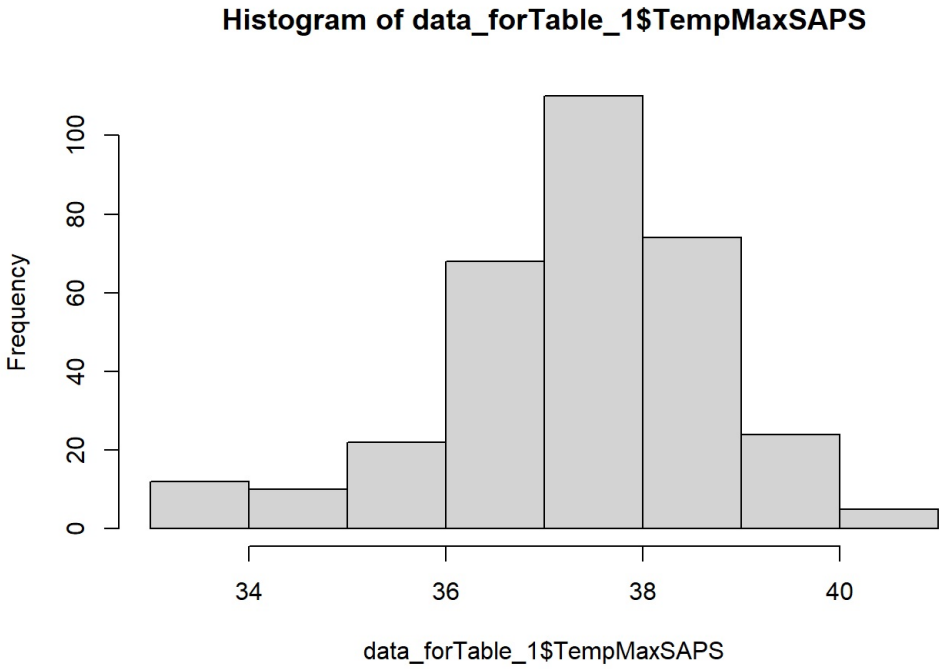

```
hist(data_forTable_1$PlateletsMinSAPS)
```



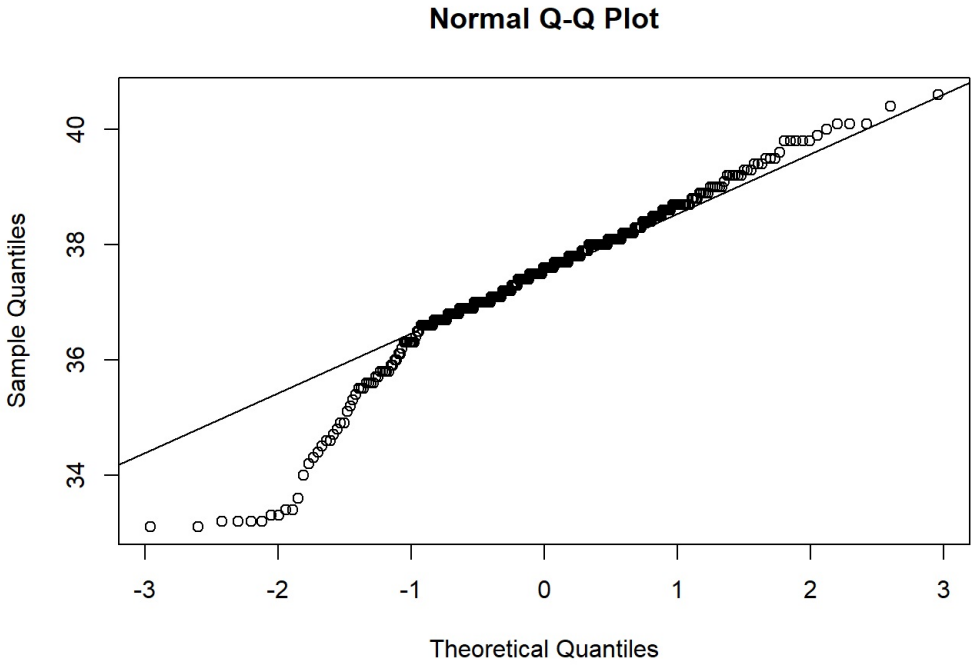

```
qqnorm(data_forTable_1$PlateletsMinSAPS); qqline(data_forTable_1$PlateletsMinSAPS) # normal?
```



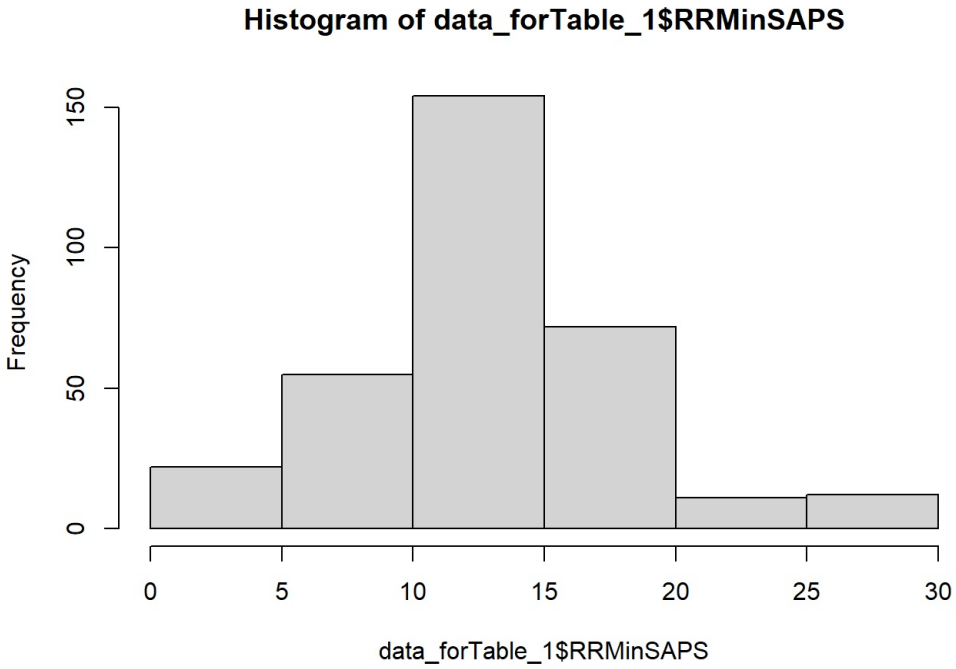

```
hist(data_forTable_1$TempMaxSAPS)
```



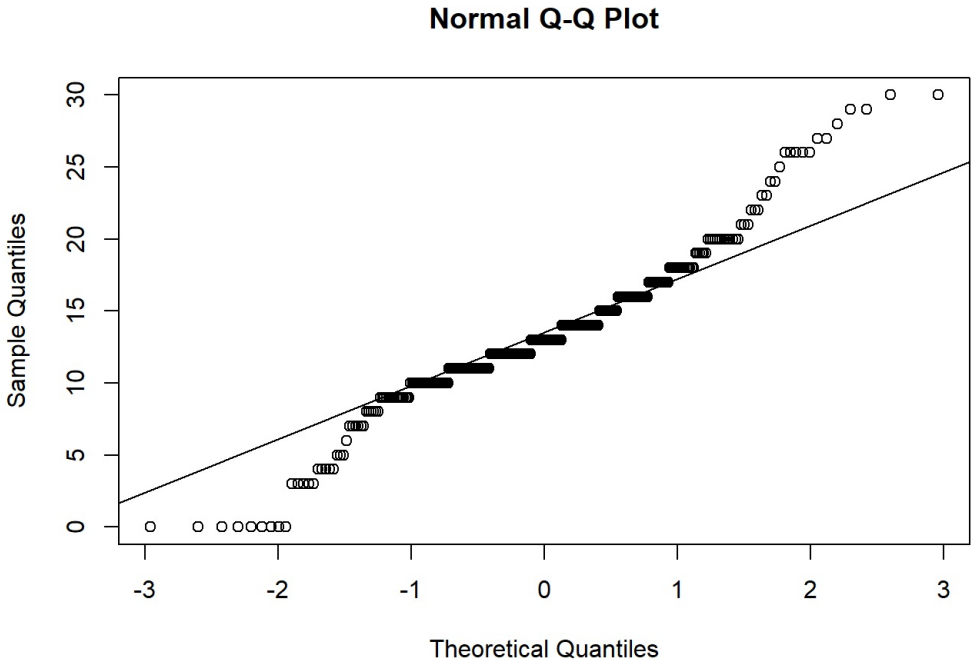

```
qqnorm(data_forTable_1$TempMaxSAPS); qqline(data_forTable_1$TempMaxSAPS) # non-normal
```



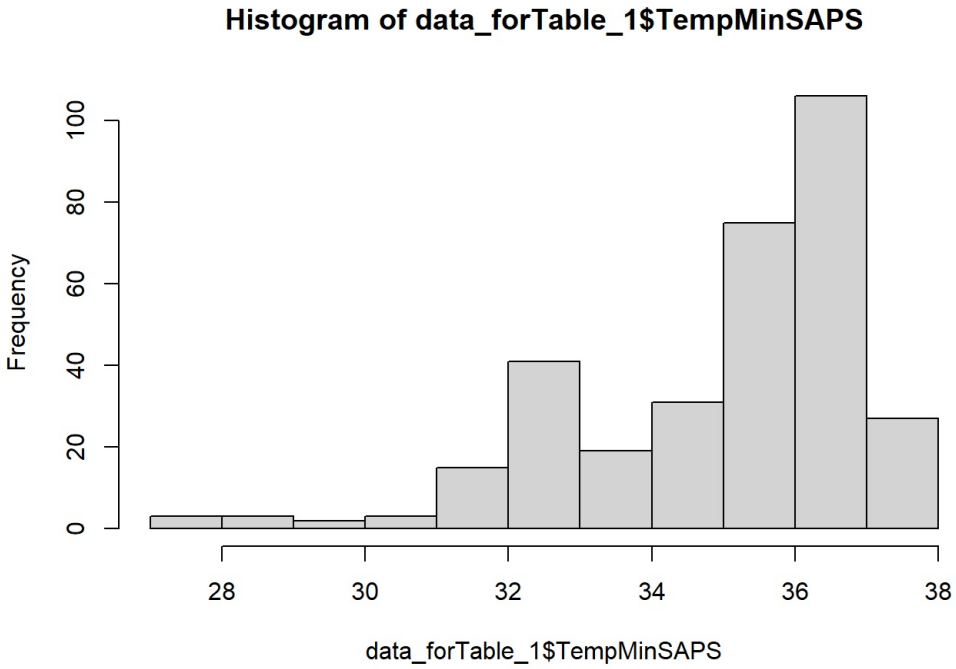

```
hist(data_forTable_1$RRMinSAPS)
```



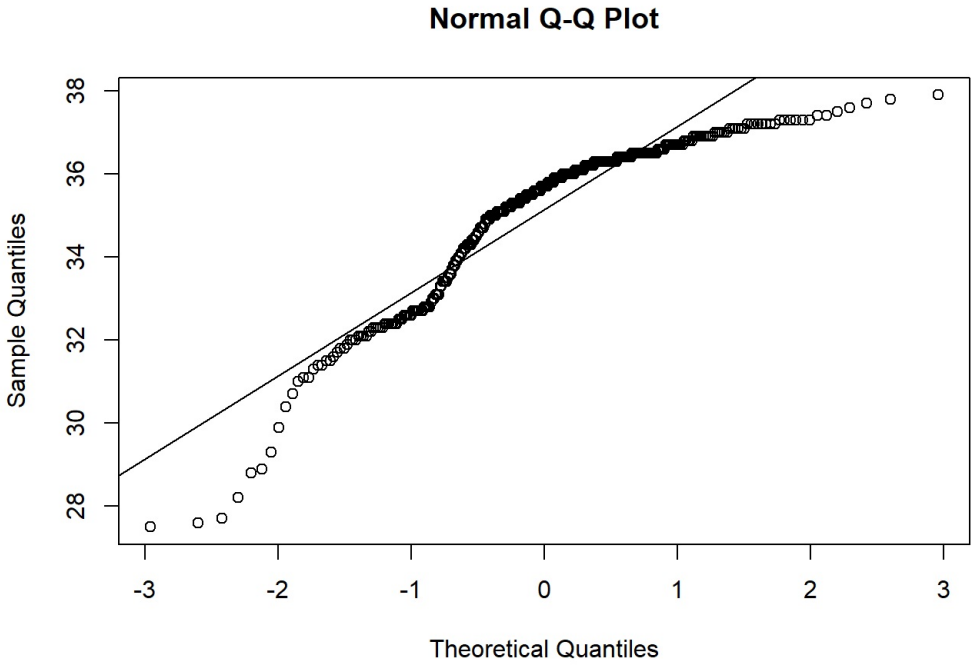

```
qqnorm(data_forTable_1$RRMinSAPS); qqline(data_forTable_1$RRMinSAPS) # non-normal
```



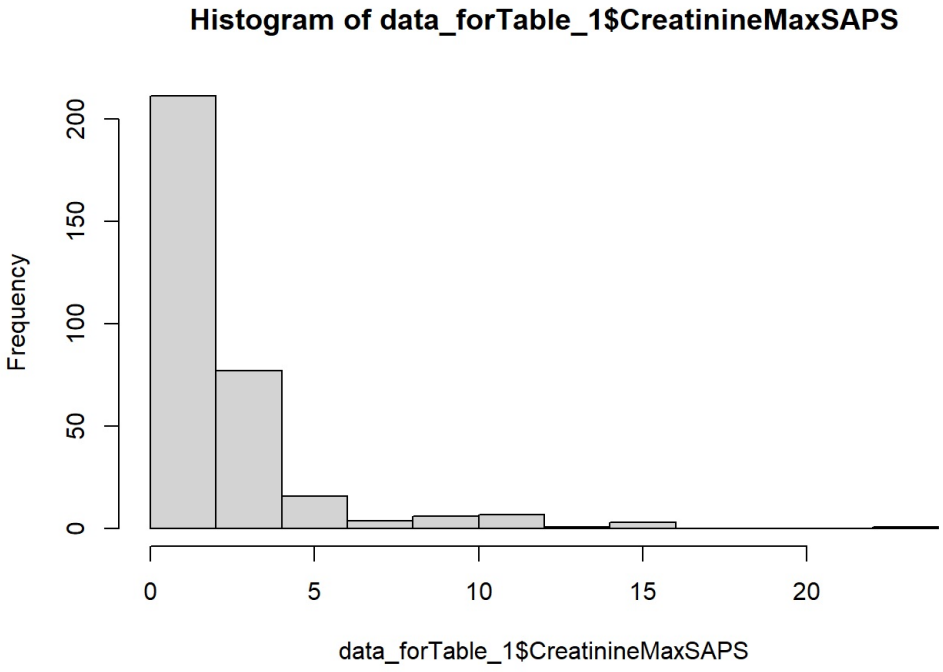

```
hist(data_forTable_1$TempMinSAPS)
```



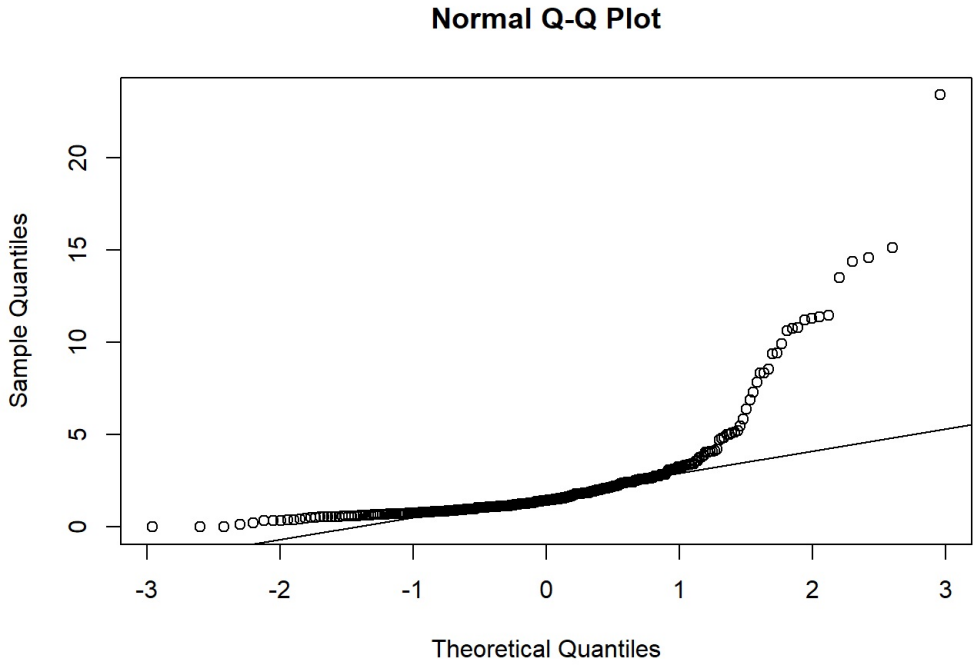

```
qqnorm(data_forTable_1$TempMinSAPS); qqline(data_forTable_1$TempMinSAPS) # non-normal
```



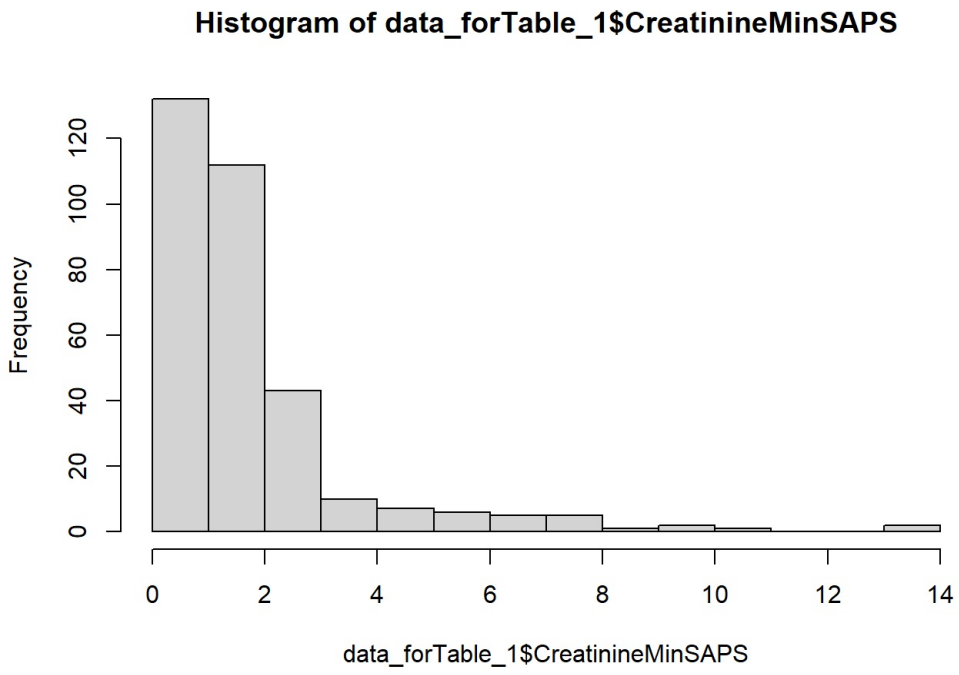

```
hist(data_forTable_1$CreatinineMaxSAPS)
```



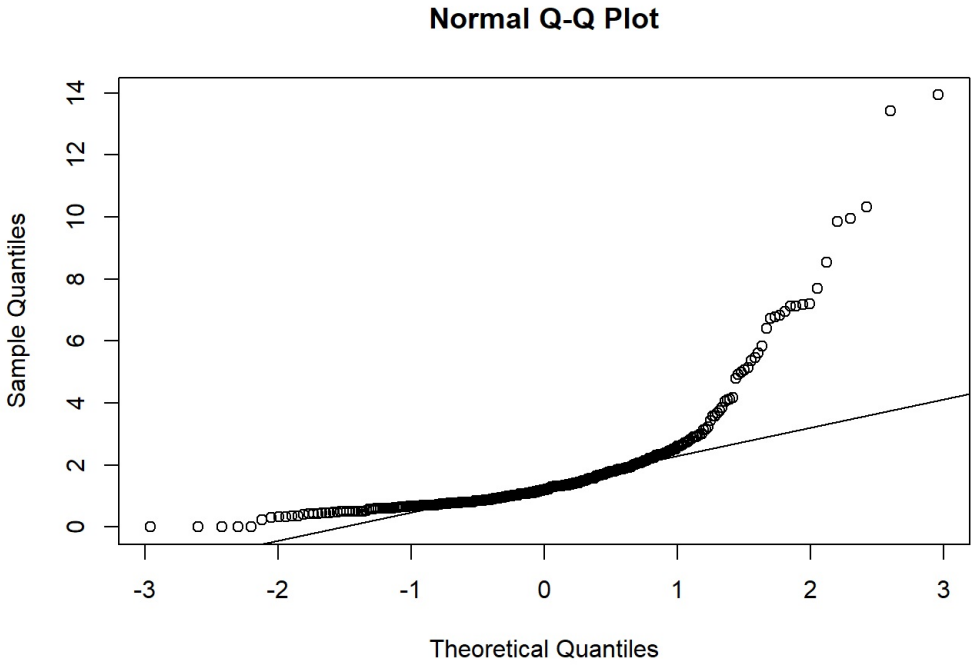

```
qqnorm(data_forTable_1$CreatinineMaxSAPS); qqline(data_forTable_1$CreatinineMaxSAPS) # non-normal
```



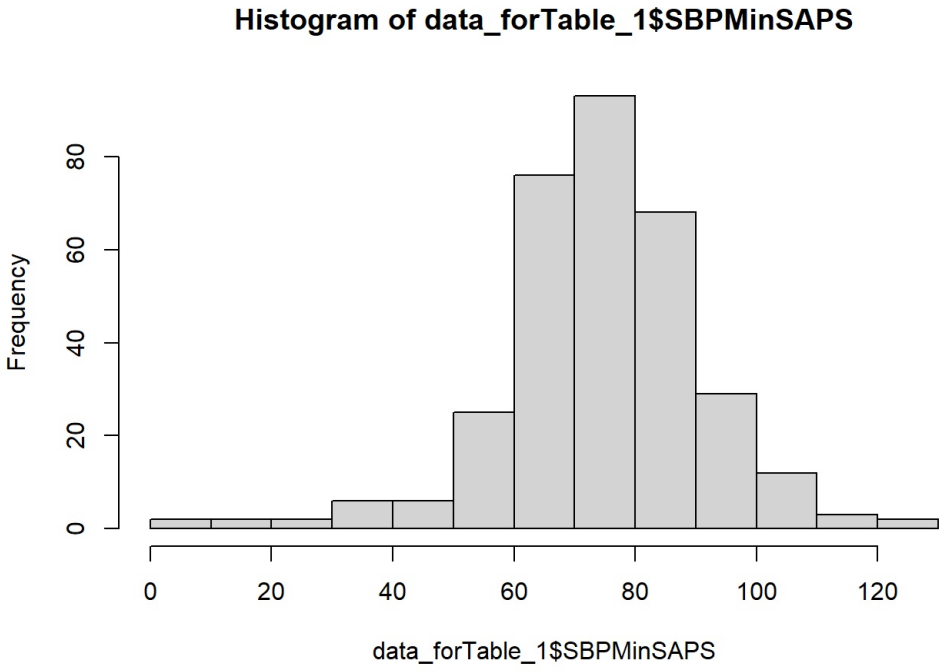

```
hist(data_forTable_1$CreatinineMinSAPS)
```



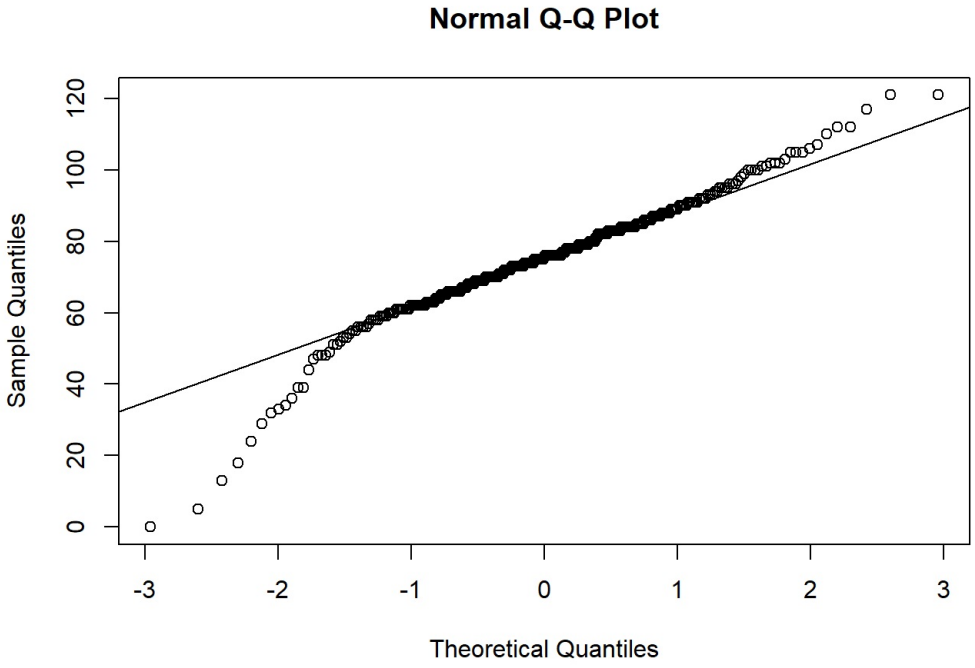

```
qqnorm(data_forTable_1$CreatinineMinSAPS); qqline(data_forTable_1$CreatinineMinSAPS) # non-normal
```



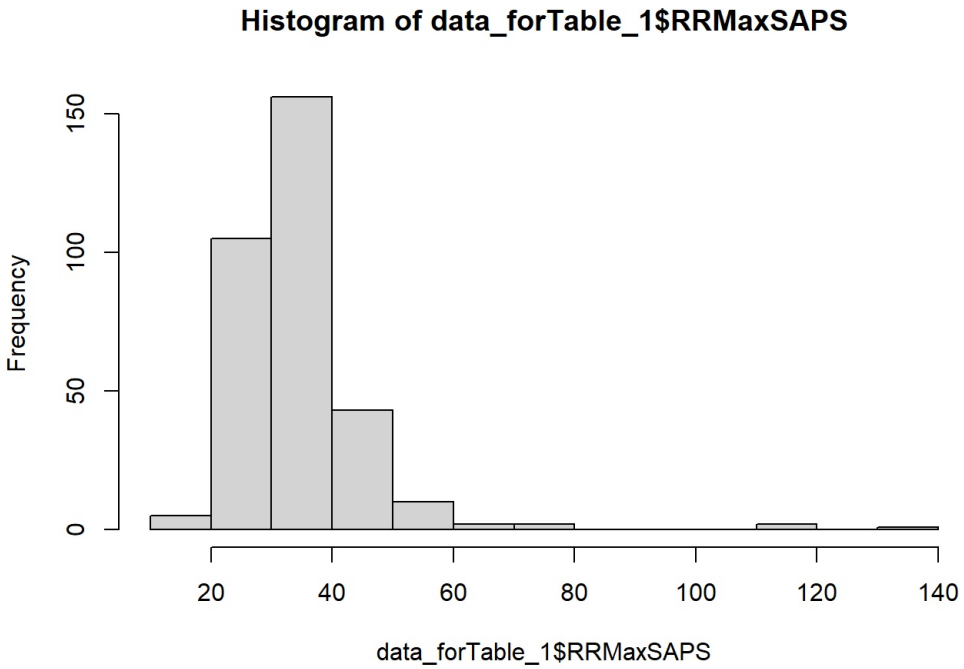

```
hist(data_forTable_1$SBPMinSAPS)
```




```
qqnorm(data_forTable_1$SBPMinSAPS); qqline(data_forTable_1$SBPMinSAPS) # non-normal
```



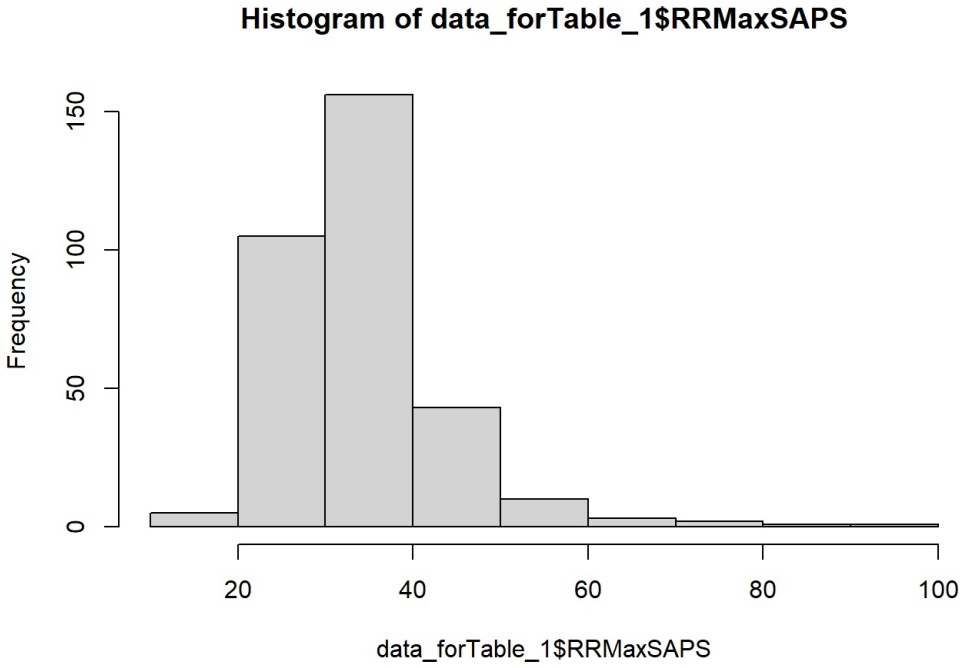

```
# fix outliers of RRMaxSAPS ####
hist(data_forTable_1$RRMaxSAPS) # there are high outliers for RRMaxSAPS
```



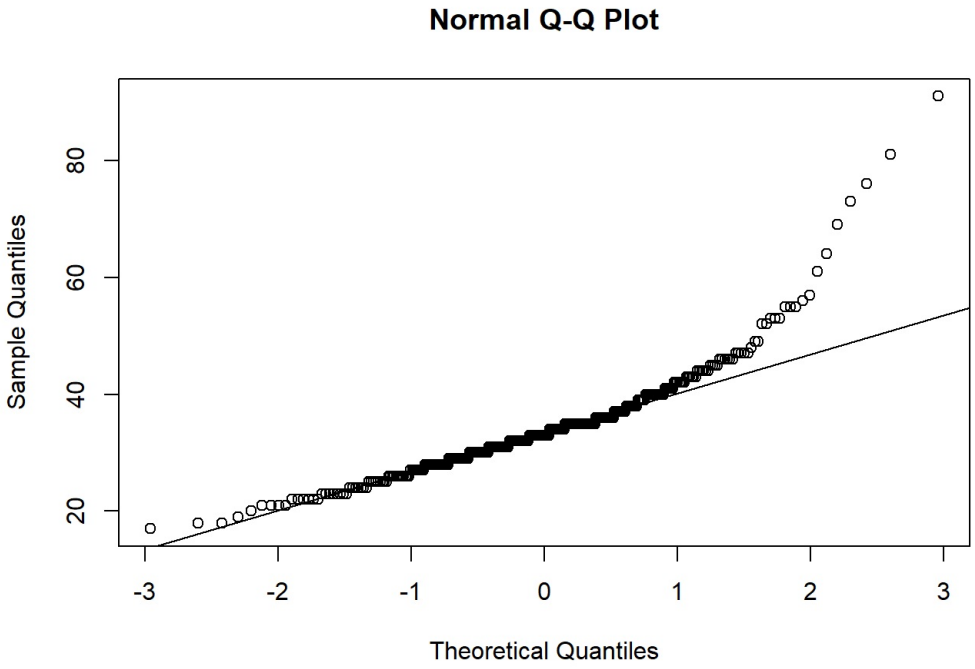

```
data_forTable_1[RRMaxSAPS>70] # EARLI IDs 1127, 1246, 1270
```

```
##    EARLIStudyId Age Gender             Race HospWithin30d PNA_community
## 1:          810  68   Male        Caucasian             0             0
## 2:          908  54 Female African American             0             0
## 3:         1127  70 Female African American             0             0
## 4:         1246  43 Female            Asian             0             0
## 5:         1270  76 Female        Caucasian             0             0
##    PNA_healthcare COPD2 CongestiveHeartFailure CardiacArrest PtSepsis
## 1:              0     0                      0             0        0
## 2:              0     1                      0             1        0
## 3:              1     0                      0             0        1
## 4:              0     0                      0             0        0
## 5:              0     1                      0             0        0
##          Group APACHEIII 28d death 60d death Hospital death MechVent
## 1: 4_NO_Sepsis        64         1         1              1        1
## 2: 4_NO_Sepsis        91         1         1              1        1
## 3: 4_NO_Sepsis        89         0         0              0        1
## 4: 4_NO_Sepsis        80         0         0              0        1
## 5:   5_Unclear        66         1         1              1        1
##    TempMaxSAPS TempMinSAPS WBCMaxSAPS WBCMinSAPS HRMaxSAPS HRMinSAPS RRMaxSAPS
## 1:          NA          NA       15.5       15.5        94        67        76
## 2:        37.3        36.3       25.2       11.9       103        46        73
## 3:        35.5        34.4       22.4       20.4       110        57       137
## 4:        36.7        31.1       51.9       31.8       138        60       117
## 5:        33.2        32.3       17.7       17.7        80        56       118
##    RRMinSAPS SIRS_HR SIRS_temp SIRS_RR SIRS_WBC SIRS_total SBPMinSAPS
## 1:        27       1         1       1        1          4         34
## 2:        12       1         0       1        1          3         59
## 3:         0       1         1       1        1          4         93
## 4:         0       1         1       1        1          4          0
## 5:        11       0         1       1        1          3        105
##    SBPMaxSAPS CreatinineMaxSAPS CreatinineMinSAPS PlateletsMinSAPS APACHEII
## 1:         95              2.24              2.24             230      -99
## 2:        229              3.88              2.62               0       44
## 3:        186              3.24              2.73             125       37
## 4:        113              0.12              0.00             253       46
## 5:        147              0.72              0.61             235       28
##    Intubated OnPressorsSAPS ER_admit_date  BirthDate virusPresent SOT HTN
## 1:         1              1    2015-01-13 1946-08-19            0   0   1
## 2:         1              1    2015-06-30 1960-10-08            0   0   1
## 3:         1              1    2016-07-20 1945-12-24            0   0   0
## 4:         1              1    2017-01-02 1973-02-01            0   0   0
## 5:         1              1    2017-01-24 1940-03-16            0   0   1
##    Cirrhosis CKD Malignancy Immunocompromised HIV Diabetes
## 1:         0   1          1                 0   0        0
## 2:         0   1          0                 0   0        1
## 3:         0   1          0                 0   0        1
## 4:         0   0          0                 0   0        0
## 5:         0   0          0                 0   0        0
```

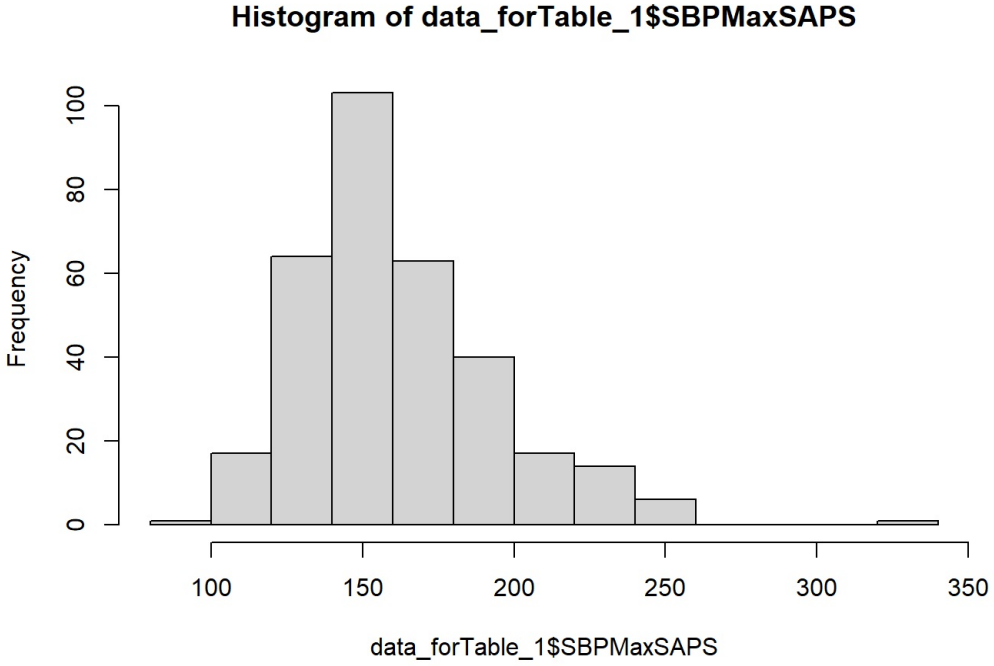
```
# Chart reviewed for EARLI ID 1127, max RR 64
data_forTable_1[EARLIStudyId==1127]$RRMaxSAPS <- 64

# Chart reviewed for EARLI ID 1246, max RR 91
data_forTable_1[EARLIStudyId==1246]$RRMaxSAPS <- 91

# Chart reviewed for EARLI ID 1270, max RR 81
data_forTable_1[EARLIStudyId==1270]$RRMaxSAPS <- 81

hist(data_forTable_1$RRMaxSAPS)
```




```
qqnorm(data_forTable_1$RRMaxSAPS); qqline(data_forTable_1$RRMaxSAPS) # non-normal
```



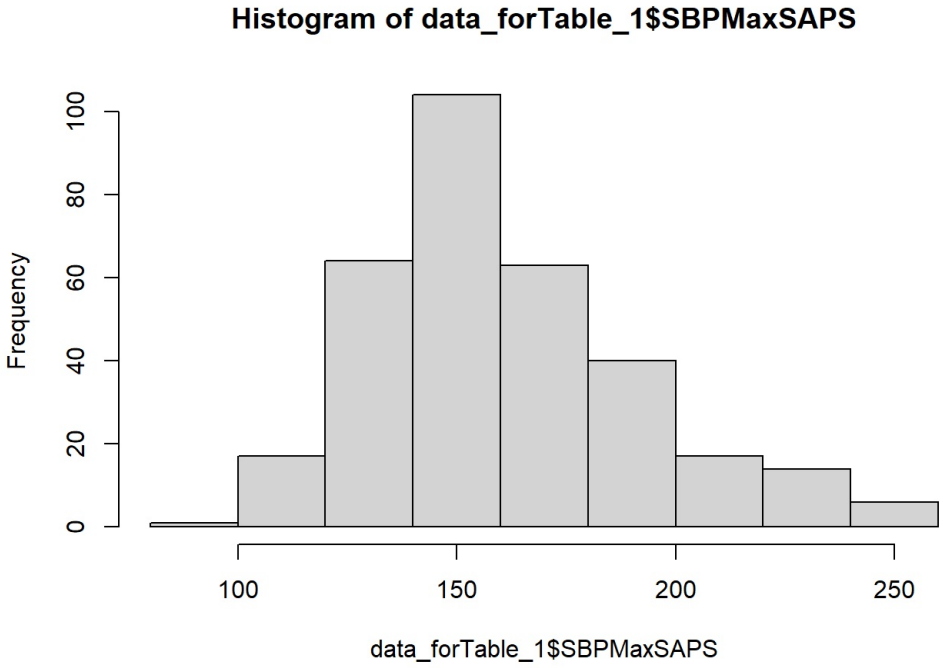

```
# fix outliers for SBPMaxSAPS ####
hist(data_forTable_1$SBPMaxSAPS) # there is a high outlier for SBPMaxSAPS, normal
```



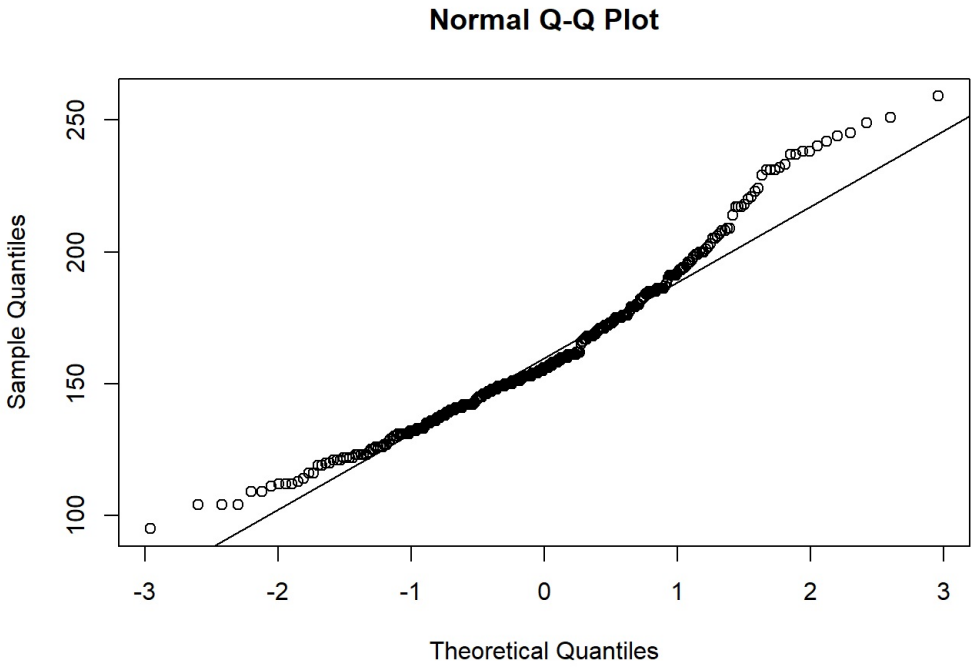

```
data_forTable_1[SBPMaxSAPS>300] # EARLI ID 598
```

```
##    EARLIStudyId Age Gender  Race HospWithin30d PNA_community PNA_healthcare
## 1:          598  66   Male Asian             1             0              0
##    COPD2 CongestiveHeartFailure CardiacArrest PtSepsis       Group APACHEIII
## 1:     0                     0             0        0 4_NO_Sepsis        75
##    28d death 60d death Hospital death MechVent TempMaxSAPS TempMinSAPS
## 1:         1         1              1        1        33.3        32.3
##    WBCMaxSAPS WBCMinSAPS HRMaxSAPS HRMinSAPS RRMaxSAPS RRMinSAPS SIRS_HR
## 1:        7.6        5.8       103        85        32         8       1
##    SIRS_temp SIRS_RR SIRS_WBC SIRS_total SBPMinSAPS SBPMaxSAPS
## 1:         1       1        0          3         82        330
##    CreatinineMaxSAPS CreatinineMinSAPS PlateletsMinSAPS APACHEII Intubated
## 1:              2.35              1.81              133       40         1
##    OnPressorsSAPS ER_admit_date  BirthDate virusPresent SOT HTN Cirrhosis CKD
## 1:              1    2014-01-06 1947-04-07            0   1   1         0   1
##    Malignancy Immunocompromised HIV Diabetes
## 1:          0                 1   0        0
```

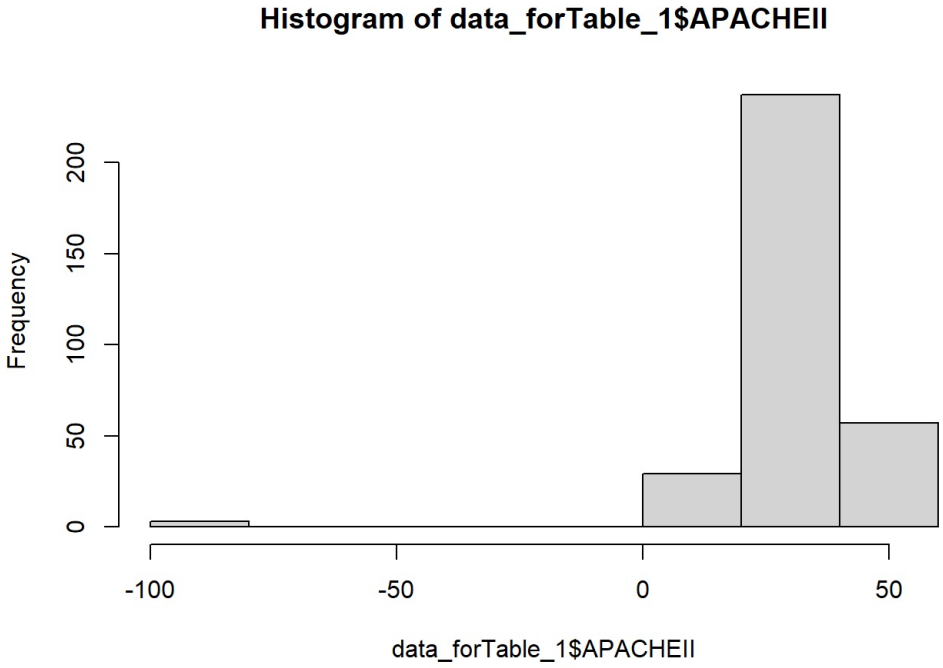
```
# Chart reviewed for EARLI ID 598, max SBP 155
data_forTable_1[EARLIStudyId==598]$SBPMaxSAPS <- 155

hist(data_forTable_1$SBPMaxSAPS)
```




```
qqnorm(data_forTable_1$SBPMaxSAPS); qqline(data_forTable_1$SBPMaxSAPS) # normal?
```



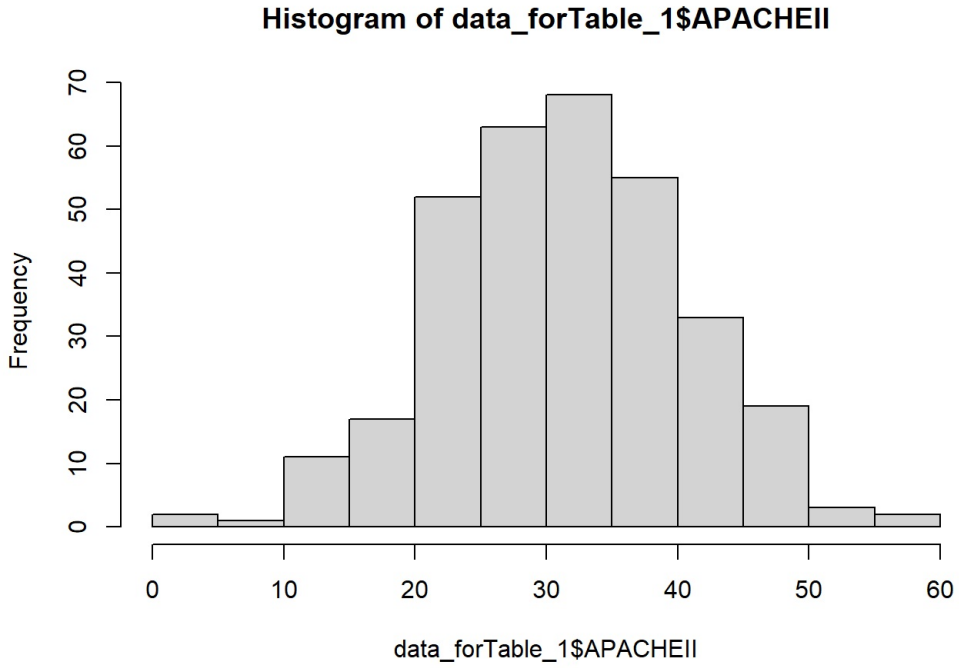

```
# fix outliers for APACHEII
hist(data_forTable_1$APACHEII) # there is a negative APACHE; otherwise, normal
```



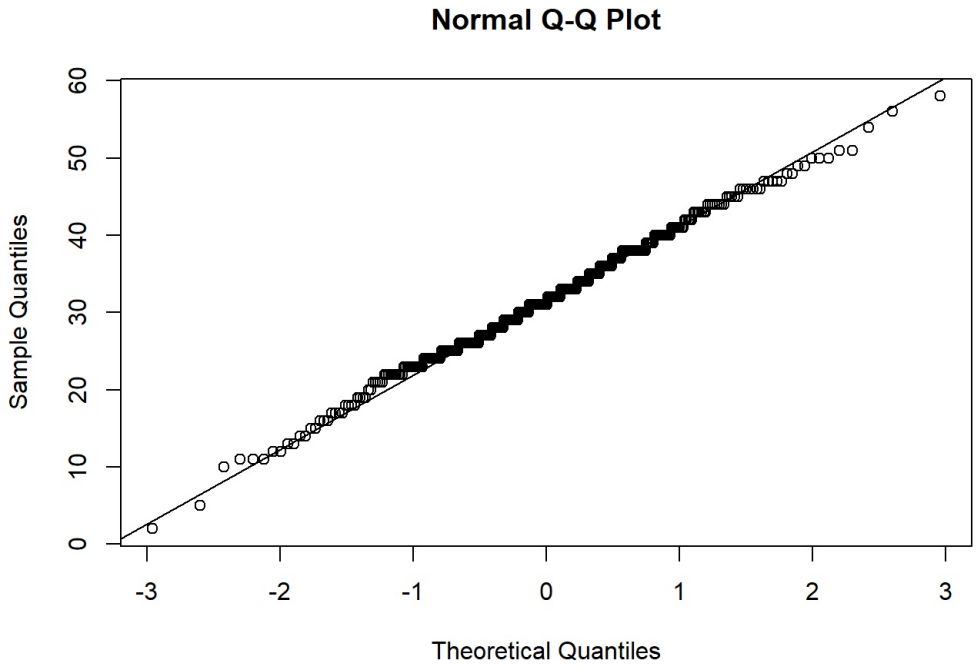

```
data_forTable_1[APACHEII<0] # EARLI ID 810, 1095, 1255
```

```
##    EARLIStudyId Age Gender      Race HospWithin30d PNA_community PNA_healthcare
## 1:          810  68   Male Caucasian             0             0              0
## 2:         1095  66   Male Caucasian             0             0              0
## 3:         1255  43   Male     Asian             1             0              0
##    COPD2 CongestiveHeartFailure CardiacArrest PtSepsis       Group APACHEIII
## 1:     0                     0             0        0 4_NO_Sepsis        64
## 2:     0                     1             0        0 4_NO_Sepsis        51
## 3:     0                     1             0        0 4_NO_Sepsis        22
##    28d death 60d death Hospital death MechVent TempMaxSAPS TempMinSAPS
## 1:         1         1              1        1          NA          NA
## 2:         0         0              0        0        36.8        36.5
## 3:         0         0              0        1        37.5        37.1
##    WBCMaxSAPS WBCMinSAPS HRMaxSAPS HRMinSAPS RRMaxSAPS RRMinSAPS SIRS_HR
## 1:       15.5       15.5        94        67        76        27       1
## 2:        9.5        6.5       108        73        32        17       1
## 3:       11.8       11.8       109        81        28        10       1
##    SIRS_temp SIRS_RR SIRS_WBC SIRS_total SBPMinSAPS SBPMaxSAPS
## 1:         1       1        1          4         34         95
## 2:         0       1        0          2         77        151
## 3:         0       1        0          2         84        136
##    CreatinineMaxSAPS CreatinineMinSAPS PlateletsMinSAPS APACHEII Intubated
## 1:              2.24              2.24              230      -99         1
## 2:              0.64              0.50              126      -99         0
## 3:              0.00              0.00              235      -99         1
##    OnPressorsSAPS ER_admit_date  BirthDate virusPresent SOT HTN Cirrhosis CKD
## 1:              1    2015-01-13 1946-08-19            0   0   1         0   1
## 2:              0    2016-06-13 1950-03-01            0   0   0         0   0
## 3:              0    2017-01-09 1973-11-30            0   0   1         0   0
##    Malignancy Immunocompromised HIV Diabetes
## 1:          1                 0   0        0
## 2:          0                 0   0        0
## 3:          0                 0   0        0
```

```
# Chart reviewed for EARLI ID 810, APACHE could not be calculated because temp missing
# Not febrile, assumed temp was 37C for calcalation
data_forTable_1[EARLIStudyId==810]$APACHEII <- 24

# Chart reviewed for EARLI ID 1095 and APACHEII calculated
data_forTable_1[EARLIStudyId==1095]$APACHEII <- 5

# Chart reviewed for EARLI ID 1255 and APACHEII calculated
data_forTable_1[EARLIStudyId==1255]$APACHEII <- 2

hist(data_forTable_1$APACHEII)
```




```
qqnorm(data_forTable_1$APACHEII); qqline(data_forTable_1$APACHEII) # normal
```



```
# Create TableOne object variables
myVars <- c("Age", "APACHEII", "APACHEIII", "TempMaxSAPS", "TempMinSAPS", "WBCMaxSAPS",
            "WBCMinSAPS", "HRMaxSAPS", "HRMinSAPS", "RRMaxSAPS", "RRMinSAPS", "SBPMaxSAPS",
            "SBPMinSAPS", "CreatinineMaxSAPS", "CreatinineMinSAPS", "PlateletsMinSAPS",
            "Gender", "Race", "HospWithin30d", "PNA_community", "PNA_healthcare",
            "COPD2", "CongestiveHeartFailure", "CardiacArrest", "PtSepsis",
            "28d death", "60d death", "Hospital death", "MechVent", "SIRS_HR",
            "SIRS_temp", "SIRS_RR", "SIRS_WBC", "SIRS_total", "Intubated",
            "OnPressorsSAPS", "virusPresent", "Immunocompromised", "SOT", "HTN",
            "Cirrhosis", "CKD", "Malignancy", "HIV", "Diabetes")

myFactorVars <- c("Gender", "Race", "HospWithin30d", "PNA_community", "PNA_healthcare",
                  "COPD2", "CongestiveHeartFailure", "CardiacArrest", "PtSepsis",
                  "28d death", "60d death", "Hospital death", "MechVent", "SIRS_HR",
                  "SIRS_temp", "SIRS_RR", "SIRS_WBC", "SIRS_total", "Intubated",
                  "OnPressorsSAPS", "virusPresent", "Immunocompromised", "SOT", "HTN",
                  "Cirrhosis", "CKD", "Malignancy", "HIV", "Diabetes")

# Create the print function variables
myNonnormals <- c("TempMaxSAPS", "RRMinSAPS", "TempMinSAPS", "CreatinineMaxSAPS",
                  "CreatinineMinSAPS", "SBPMinSAPS", "RRMaxSAPS")

# unsure if normal
# when treated as non-normal
### Age becomes non-significant
### HRMin and HRMax remain highly significant
### Platelets remains highly significant
### SBPMax becomes significant
myNonnormals_more <- c(myNonnormals, "Age", "HRMinSAPS", "HRMaxSAPS",
                       "PlateletsMinSAPS")

myExact <- c("Caucasian", "African American", "Native American", "Asian", "Other",
             "Unknown", "PNA_healthcare", "PNA_community", "CardiacArrest",
             "PtSepsis", "1", "2", "3", "4", "virusPresent","Immunocompromised",
             "SOT", "Cirrhosis", "HIV")

table1 <- CreateTableOne(vars = myVars, data = data_forTable_1, strata = "Group",
                                factorVars = myFactorVars)
```