Linear Regression Models: Final Project

Michael Demian – md3726

Introduction:

For this final project as we have a scientific goal of determining which mutations of HIV-1 are associated with drug resistance, we will test different models and their subsequent sets of mutation positions to determine which is the best predictor with an alpha of 0.05. Given our goal, we would like to identify all possible mutations of HIV-1 that are associated with drug resistance, as well as avoiding falsely identifying a mutation position that is, in fact, not associated with drug resistance. Weighting between identifying the maximum number of mutations and the minimum number of false discoveries can be done in many different ways, but any ideal should have all possible true mutations identified and no false mutations (i.e. position list should match that of tsm_df positions). In our evaluation we will present the data for the methods and then choose with best being the maximum that satisfies (# of true discoveries/# of all possible true mutations) - false discovery rate across all 7 drugs. We are not only using false discovery rate as our tool as we believe that a method that produces something such as 1 true discovery and 0 false discoveries is worse than a method producing 9 true discoveries and 1 false discovery.

Background on Methods Used:

There will be seven main methods tested against each other from 2 different classes. After the data has been cleaned and correlation effects dealt with (work done by professor), we will be starting with a marginal T-test to produce the p-values. In this project began by starting with Bonferroni and comparing that to just anything with marginal under alpha, came across variations of Bonferroni (those by Holm, Hochberg, and Hommel) within the Family Wise Error Rate methods. Also came across False Discovery Rate methods with Benjamini-Hochberg (BH) and Benjamini-Yekutieli (BY). These six methods, along with a simple pass through, will all be tested after the data is cleaned, processed, and accounted for collinearity to estimate what variables will be kept. I believe the Marginal T-test along with Bonferroni has been described a lot previously, I will give brief definitions of the other five.

Family Wise Error Rates:

Holm Correction - "Begin by ordering the k hypotheses by their respective p-values. Select the lowest p-value and compare it to α/k. If the p-value is lower, then [include that predictor] and perform the same selection with the remaining $k-1$ [predictors] and a threshold of $\alpha/(k-1)$. Repeat the process until the selected p-value is not smaller, at which time all remaining [predictors] should be [excluded]."

Hochberg Correction - Very similar start to the Holm correction, "However, Hochberg method conducts statistical inference of hypothesis by starting with the largest P value (H(m), …, H(1)). When we first observe $p(i) < \alpha'(i)$ for hypothesis H(i) (i=m,…,1), the comparison stops and then concludes that the hypotheses of H(j) (j=i,…,1) will be rejected at significance level α. It is also known that Hochberg adjustment is more powerful than Holm method."

Hommel Correction - "Simes (1986) modified Bonferroni method and proposed a global test of m hypotheses (8). Let H={ H(1), …, H(m)} be the global intersection hypothesis, H will be rejected if p(i) ≤ iα/m for any i=1, ..., m. However, Simes global test could not be used for assessing the individual hypothesis Hi. Therefore, Hommel (1988) extended Simes' method for testing individual Hi (9). Let an index of $j = maxi \in 1, \ldots, m{:}\, p(m - i + k) > k\alpha/i$ for $k = 1, \ldots, i$ be the size of the largest subset of m hypotheses for which Simes test is not significant. All Hi (i=1,...,m) are rejected if j does not exist, otherwise reject all Hi with pi≤α/j"

False Discovery Rates:

Benjamini-Hochberg - "In contrast to the strong control of FWER, Benjamini and Hochberg [1995] introduced a method for controlling FDR, which is herein termed BH adjustment (10). Let be the pre-specified upper bound of FDR (e.g., q=0.05), the first step is to compute index:

$$k = max(i{:}\, p(i) \leq \frac{i}{m}q)$$

If k does not exist, reject no hypothesis, otherwise reject hypothesis of Hi (i=1,...,k). BH method starts with comparing H(i) from the largest to smallest P value (i=m,...,1). The FDR-based control is less stringent with the increased gain in power and has been widely used in cases where a large number of hypotheses are simultaneously tested."

Benjamini-Yekutieli - Very similar to the above model, with a correction on the q such that:

$$k = max(i{:}\, p(i) \leq \frac{i}{m}q^{\sim})\text{: where, } q^{\sim} = \frac{q}{\sum_{i=1}^{m}\frac{1}{i}}$$

Analysis of HIV Drug Resistance Data

The scientific goal is to determine which mutations of the Human Immunodeficiency Virus Type 1 (HIV-1) are associated with drug resistance. The data set, publicly available from the Stanford HIV Drug Resistance Database http://hivdb.stanford.edu/pages/published_analysis/genophenoPNAS2006/, was originally analyzed in (Rhee et al. 2006).

Preparing the data

The data set consists of measurements for three classes of drugs: protease inhibitors (PIs), nucleoside reverse transcriptase (RT) inhibitors (NRTIs), and nonnucleoside RT inhibitors (NNRTIs). Protease and reverse transcriptase are two enzymes in HIV-1 that are crucial to the function of the virus. This data set seeks associations between mutations in the HIV-1 protease and drug resistance to different PI type drugs, and between mutations in the HIV-1 reverse transcriptase and drug resistance to different NRTI and NNRTI type drugs (The raw data are saved as `gene_df`).

In order to evaluate our results, we compare with the treatment-selected mutation panels created by (Rhee et al. 2005), which can be viewed as the ground true. These panels give lists of HIV-1 mutations appearing more frequently in patients who have previously been treated with PI, NRTI, or NNRTI type drugs, than in patients with no previous exposure to that drug type. Increased frequency of a mutation among patients treated with a certain drug type implies that the mutation confers resistance to that drug type (The raw data are saved as `tsm_df`).

To simplify the analysis, in this project we will confine our attention to the PI drugs.

```
drug_class = 'PI' # Possible drug types are 'PI', 'NRTI', and 'NNRTI'.
```

Fetching and cleaning the data

First, we download the data and read it into data frames.

```
base_url = 'http://hivdb.stanford.edu/pages/published_analysis/genophenoPNAS2006'
gene_url = paste(base_url, 'DATA', paste0(drug_class, '_DATA.txt'), sep='/')
tsm_url = paste(base_url, 'MUTATIONLISTS', 'NP_TSM', drug_class, sep='/')

gene_df = read.delim(gene_url, na.string = c('NA', ''), stringsAsFactors = FALSE)
tsm_df = read.delim(tsm_url, header = FALSE, stringsAsFactors = FALSE)
names(tsm_df) = c('Position', 'Mutations')
```

A small sample of the data is shown below.

```
head(gene_df, n=6)

##   IsolateName PseudoName MedlineID  APV  ATV   IDV   LPV   NFV   RTV   SQV
## 1     CA10676      CA622  10839657  2.3   NA  32.7    NA  23.4  51.6  37.8
## 2     CA37880      CA622  15995959 76.0   NA 131.0 200.0  50.0 200.0 156.0
## 3      CA9984      CA624  11897594  2.8   NA  12.0    NA 100.0  41.0 145.6
## 4     CA17003      CA628  15995959  6.5  9.2   2.1   5.3   5.0  36.0  13.0
## 5     CA10670      CA634  10839657  8.3   NA 100.0    NA 161.1 170.2 100.0
## 6     CA42683      CA634  15995959 82.0 75.0 400.0 400.0  91.0 400.0 400.0
##   P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20
## 1  -  -  -  -  -  -  -  -  -   I   -   -   -   -   -   -   -   -   -   -
## 2  -  -  -  -  -  -  -  -  -   F   L   -   -   -   -   A   -   -   I   -
## 3  -  -  -  -  -  -  -  -  -   -   -   -   V   -   -   -   -   -   -   R
## 4  -  -  -  -  -  -  -  -  -   I   -   -   -   -   -   -   -   -   -   -
## 5  -  -  -  -  -  -  -  -  -   I   -   -   -   -   -   -   -   -   -   R
## 6  -  -  -  -  -  -  -  -  -   I   -   -   -   -   -   -   -   -   -   R
##   P21 P22 P23 P24 P25 P26 P27 P28 P29 P30 P31 P32 P33 P34 P35 P36 P37 P38
## 1   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -  DN   -
## 2   -   -   -   -   -   -   -   -   -   -   -   -   F   -   -   -   -   -
## 3   -   -   -   -   -   -   -   -   -   N   -   -  IL   -   D   I   -   -
## 4   -   -   I   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -
## 5   -   -   F   -   -   -   -   -   -   -   -   -   -   -   D   I   D   -
## 6   -   -   -   -   -   -   -   -   -   -   -   -   F   T   D   I   D   -
##   P39 P40 P41 P42 P43 P44 P45 P46 P47 P48 P49 P50 P51 P52 P53 P54 P55 P56
## 1   -   -   K   -   -   -  RK   I   -   -   -   -   -   -   -   -   -   -
## 2   -   -   K   -   -   -   -   I   -   -   -   -   -   -   -   V   -   -
## 3   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   V   -   -
## 4   -   -   -   -   -   -   -   L   -   -   -   -   -   -   -   -   -   -
## 5   -   -   -   -   -   -   -   I   -   V   -   -   -   -   -   T   -   -
## 6   -   -   -   -   T   -   -   I   -   V   -   V   -   -   -   S   -   -
##   P57 P58 P59 P60 P61 P62 P63 P64 P65 P66 P67 P68 P69 P70 P71 P72 P73 P74
## 1   -   -   -   -   -   -   P   -   -   -   -   -   -   -  LV   -   S   -
## 2   -   -   -   -   -   -   P   -   -   -   -   -   -   -   V   -   S   -
## 3   -   -   -   -   -   -   P   -   -   -   -   -   -   -  TA   -   -   -
## 4   -   E   -   E  EK   -   P   -   -   -   -   -   -   -   T   T   -   -
## 5   -   -   -   -   -   V   P   -   -   -   -   -   -   -   V   X   -   -
## 6   -   -   -   -   -   V   P   -   -   -   -   -   -   -   V   V   -   -
##   P75 P76 P77 P78 P79 P80 P81 P82 P83 P84 P85 P86 P87 P88 P89 P90 P91 P92
## 1   -   -   I   -   -   -   -   T   -   V   V   -   -   -   -   M   -   -
## 2   -   -   -   -   -   -   -   T   -   V   -   -   -   -   V   M   S   -
## 3   -   -   -   -   -   -   -   -   -   -   -   -   -   D   -   M   -   -
## 4   -   -   -   -   -   -   -   A   -   V   -   -   -   -   -   M   -   -
## 5   -   -   I   -   -   -   -   A   -   -   -   -   -   -   -   -   -   -
## 6   -   -   I   -   -   -   -   A   -   -   V   -   -   -   -   -   -   -
##   P93 P94 P95 P96 P97 P98 P99
## 1   L   -   -   -   -   -   -
## 2   L   -   -   -   -   -   -
## 3   -   -   -   -   -   -   .
## 4   -   -   -   -   -   -   -
## 5   L   -   -   -   -   -   -
## 6   L   -   -   -   -   -   -
```

```r
head(tsm_df, n=6)
```

```
##   Position Mutations
## 1       10       F R
## 2       11         I
## 3       20     I T V
## 4       23         I
## 5       24         I
## 6       30         N
```

In `tsm_df`, the variable `Position` denotes the position of the mutations that are associated with drug-resistance, while `Mutations` indicating the mutation type.

The gene data table has some rows with error flags or nonstandard mutation codes. For simplicity, we remove all such rows.

```r
# Returns rows for which every column matches the given regular expression.
grepl_rows <- function(pattern, df) {
  cell_matches = apply(df, c(1,2), function(x) grepl(pattern, x))
  apply(cell_matches, 1, all)
}

pos_start = which(names(gene_df) == 'P1')
pos_cols = seq.int(pos_start, ncol(gene_df))
valid_rows = grepl_rows('^(\\.|-|[A-Zid]+)$', gene_df[,pos_cols])
gene_df = gene_df[valid_rows,]
```

Preparing the regression matrix

We now construct the design matrix X and matrix of response vectors Y. The features (columns of X) are given by mutation/position pairs. Define

$$ X_{i,j} = 1 i j \

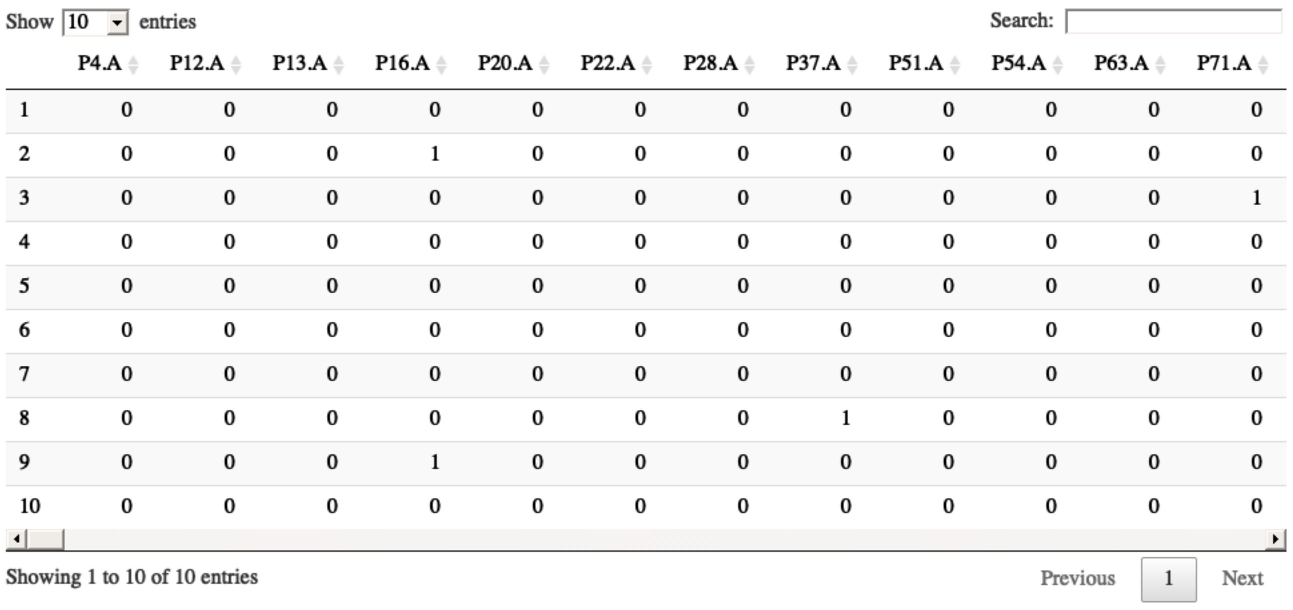
Y_{i,k} = i k. $$

For example, in the sample for PI type drugs, three different mutations (A, C, and D) are observed at position 63 in the protease, and so three columns of X (named P63.A, P63.C, and P63.D) indicate the presence or absence of each mutation at this position.

```r
# Flatten a matrix to a vector with names from concatenating row/column names.
flatten_matrix <- function(M, sep='.') {
  x <- c(M)
  names(x) <- c(outer(rownames(M), colnames(M),
                      function(...) paste(..., sep=sep)))
  x
}

# Construct preliminary design matrix.
muts = c(LETTERS, 'i', 'd')
X = outer(muts, as.matrix(gene_df[,pos_cols]), Vectorize(grepl))
X = aperm(X, c(2,3,1))
dimnames(X)[[3]] <- muts
X = t(apply(X, 1, flatten_matrix))
mode(X) <- 'numeric'

# Remove any mutation/position pairs that never appear in the data.
X = X[,colSums(X) != 0]

# Extract response matrix.
Y = gene_df[,4:(pos_start-1)]
```

An excerpt of the design matrix is shown below. By construction, every column contains at least one 1, but the matrix is still quite sparse with the relative frequency of 1's being about 0.025.

```r
library(DT)

datatable(data.frame(X)[1:10, ], options = list(scrollX=T, pageLength = 10))
```



Show 10 entries   Search:

	P4.A	P12.A	P13.A	P16.A	P20.A	P22.A	P28.A	P37.A	P51.A	P54.A	P63.A	P71.A
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	1	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	1
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	1	0	0	0	0
9	0	0	0	1	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0

Showing 1 to 10 of 10 entries   Previous 1 Next

The response matrix looks like:

```
head(Y, n=6)
```

```
##    APV  ATV   IDV   LPV   NFV   RTV   SQV
## 1  2.3   NA  32.7    NA  23.4  51.6  37.8
## 2 76.0   NA 131.0 200.0  50.0 200.0 156.0
## 3  2.8   NA  12.0    NA 100.0  41.0 145.6
## 4  6.5  9.2   2.1   5.3   5.0  36.0  13.0
## 5  8.3   NA 100.0    NA 161.1 170.2 100.0
## 6 82.0 75.0 400.0 400.0  91.0 400.0 400.0
```

There are 7 PI-type drugs: APV, ATV, IDV, LPV, NFV, RTV, and SQV.

Selecting drug-resistance-associated mutations

In this step, you need to build an appropriate linear regression model, and use the method we discussed in lecture to select mutations that may associated with drug-resistance. For 7 PI-type drugs, you need to run a seperate analysis for each drug.

Notice that there are some missing values.

Before building the model, we need to perform some final pre-processing steps. We remove rows with missing values (which vary from drug to drug) and we then further reduce the design matrix by removing predictor columns for mutations that do not appear at least three times in the sample. Finally, for identifiability, we remove any columns that are duplicates (i.e. two mutations that appear only in tandem, and therefore we cannot distinguish between their effects on the response).

```
selection <- function (X, y, fdr, alpha) {
  # Remove patients with missing measurements.
  missing = is.na(y)
  y = y[!missing]
  X = X[!missing,]

  # Remove predictors that appear less than 3 times.
  X = X[,colSums(X) >= 3]


  # Remove duplicate predictors.
  X = X[,colSums(abs(cor(X)-1) < 1e-4) == 1]

  X2 = coef(summary(lm(y~X-1)))[,4]

  #adjusted P-Values
  p = X2
  p.adjust.M <- p.adjust.methods[p.adjust.methods != "fdr"]
  p.adj    <- sapply(p.adjust.M, function(meth) p.adjust(p, meth))
  p.adj.2 <- sapply(p.adjust.M, function(meth) p.adjust(p, meth, n = length(p)))
  stopifnot(identical(p.adj[,"none"], p), p.adj <= p.adj.2)

  #Storing the different methods
  Holm = names(X2[(p.adj<alpha)[,1]])
  Hochberg = names(X2[(p.adj<alpha)[,2]])
  Hommel= names(X2[(p.adj<alpha)[,3]])
  Bonferroni= names(X2[(p.adj<alpha)[,4]])
  BH= names(X2[(p.adj<fdr)[,5]])
  BY= names(X2[(p.adj<fdr)[,6]])
  MarginalT = names(X2[(p.adj<alpha/10)[,7]])
  #final output list
  list(Holm = Holm, Hochberg = Hochberg, Hommel = Hommel,BonF = Bonferroni,BH =
BH,BY = BY, MarginalT = MarginalT)
}

alpha = 0.05 # the nominal FWER
fdr = alpha #False Detection Rate
results = lapply(Y, function(y) selection(X,y,fdr, alpha))

# results = lapply(Y, function(y) selection(X, y, fdr, alpha))
posnum <- function(x){
  sapply(regmatches(x, regexpr('[[:digit:]]+', x)), as.numeric)
}
results

## $APV
## $APV$Holm
## [1] "XP84.A" "XP84.C" "XP60.E" "XP11.I" "XP32.I" "XP11.L" "XP47.V"
##
## $APV$Hochberg
## [1] "XP84.A" "XP84.C" "XP60.E" "XP11.I" "XP32.I" "XP11.L" "XP47.V"
##
## $APV$Hommel
## [1] "XP84.A" "XP84.C" "XP60.E" "XP11.I" "XP32.I" "XP11.L" "XP47.V"
##
## $APV$BonF
## [1] "XP84.A" "XP84.C" "XP60.E" "XP11.I" "XP32.I" "XP11.L" "XP47.V"
##
## $APV$BH
##  [1] "XP84.A" "XP84.C" "XP60.E" "XP33.F" "XP11.I" "XP32.I" "XP11.L"
##  [8] "XP54.L" "XP19.P" "XP74.P" "XP34.Q" "XP20.T" "XP43.T" "XP47.V"
## [15] "XP50.V" "XP76.V" "XP89.V"
##
## $APV$BY
##  [1] "XP84.A" "XP84.C" "XP60.E" "XP11.I" "XP32.I" "XP11.L" "XP34.Q"
##  [8] "XP47.V" "XP50.V" "XP76.V" "XP89.V"
##
## $APV$MarginalT
##  [1] "XP84.A" "XP84.C" "XP60.E" "XP33.F" "XP11.I" "XP32.I" "XP11.L"
##  [8] "XP54.L" "XP19.P" "XP74.P" "XP34.Q" "XP20.T" "XP43.T" "XP47.V"
## [15] "XP50.V" "XP76.V" "XP89.V"
##
##
## $ATV
## $ATV$Holm
## [1] "XP71.I"
##
```

```
## $ATV$Hochberg
## [1] "XP71.I"
##
## $ATV$Hommel
## [1] "XP71.I"
##
## $ATV$BonF
## [1] "XP71.I"
##
## $ATV$BH
## [1] "XP82.A" "XP70.E" "XP11.I" "XP46.I" "XP71.I" "XP20.T" "XP48.V" "XP76.V"
##
## $ATV$BY
## character(0)
##
## $ATV$MarginalT
## [1] "XP82.A" "XP70.E" "XP11.I" "XP46.I" "XP71.I" "XP34.Q" "XP20.T" "XP48.V"
## [9] "XP76.V"
##
##
## $IDV
## $IDV$Holm
##  [1] "XP54.A" "XP84.A" "XP69.H" "XP57.R" "XP69.R" "XP54.S" "XP73.S"
##  [8] "XP43.T" "XP54.T" "XP48.V" "XP76.V"
##
## $IDV$Hochberg
##  [1] "XP54.A" "XP84.A" "XP69.H" "XP57.R" "XP69.R" "XP54.S" "XP73.S"
##  [8] "XP43.T" "XP54.T" "XP48.V" "XP76.V"
##
## $IDV$Hommel
##  [1] "XP54.A" "XP84.A" "XP69.H" "XP57.R" "XP69.R" "XP54.S" "XP73.S"
##  [8] "XP43.T" "XP54.T" "XP48.V" "XP76.V"
##
## $IDV$BonF
##  [1] "XP54.A" "XP84.A" "XP69.H" "XP69.R" "XP54.S" "XP73.S" "XP43.T"
##  [8] "XP54.T" "XP48.V" "XP76.V"
##
## $IDV$BH
##  [1] "XP54.A" "XP84.A" "XP69.H" "XP85.I" "XP92.K" "XP54.L" "XP19.Q"
##  [8] "XP57.R" "XP69.R" "XP54.S" "XP73.S" "XP12.T" "XP43.T" "XP54.T"
## [15] "XP20.V" "XP48.V" "XP72.V" "XP76.V" "XP67.Y"
##
## $IDV$BY
##  [1] "XP54.A" "XP84.A" "XP69.H" "XP57.R" "XP69.R" "XP54.S" "XP73.S"
##  [8] "XP12.T" "XP43.T" "XP54.T" "XP48.V" "XP76.V" "XP67.Y"
##
## $IDV$MarginalT
##  [1] "XP54.A" "XP84.A" "XP69.H" "XP85.I" "XP92.K" "XP54.L" "XP19.Q"
##  [8] "XP57.R" "XP69.R" "XP54.S" "XP73.S" "XP12.T" "XP43.T" "XP54.T"
## [15] "XP20.V" "XP48.V" "XP72.V" "XP76.V" "XP67.Y"
##
##
```

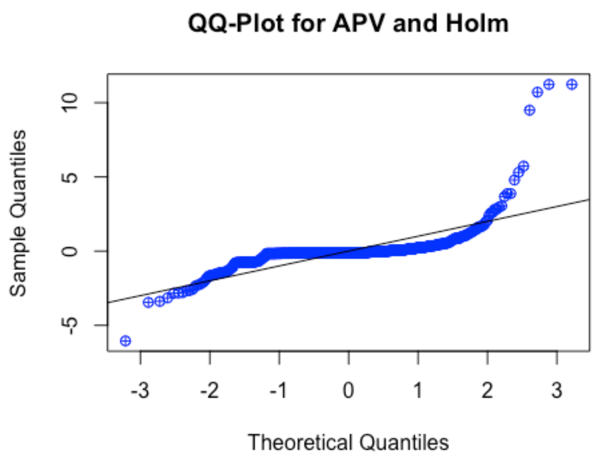
```
## $LPV
## $LPV$Holm
## [1] "XP54.A" "XP33.F" "XP69.R" "XP47.V"
##
## $LPV$Hochberg
## [1] "XP54.A" "XP33.F" "XP69.R" "XP47.V"
##
## $LPV$Hommel
## [1] "XP54.A" "XP33.F" "XP69.R" "XP47.V"
##
## $LPV$BonF
## [1] "XP54.A" "XP33.F" "XP69.R" "XP47.V"
##
## $LPV$BH
##  [1] "XP54.A" "XP33.F" "XP11.L" "XP55.R" "XP69.R" "XP54.S" "XP91.S"
##  [8] "XP43.T" "XP47.V" "XP54.V"
##
## $LPV$BY
## [1] "XP54.A" "XP33.F" "XP69.R" "XP47.V"
##
## $LPV$MarginalT
##  [1] "XP54.A" "XP58.E" "XP33.F" "XP11.L" "XP55.R" "XP69.R" "XP54.S"
##  [8] "XP91.S" "XP43.T" "XP47.V" "XP50.V" "XP54.V"
##
##
## $NFV
## $NFV$Holm
## [1] "XP82.A" "XP84.A" "XP84.C" "XP88.D" "XP30.N" "XP20.T" "XP54.T" "XP54.V"
##
## $NFV$Hochberg
## [1] "XP82.A" "XP84.A" "XP84.C" "XP88.D" "XP30.N" "XP20.T" "XP54.T" "XP54.V"
##
## $NFV$Hommel
## [1] "XP82.A" "XP84.A" "XP84.C" "XP88.D" "XP30.N" "XP20.T" "XP54.T" "XP54.V"
##
## $NFV$BonF
## [1] "XP82.A" "XP84.A" "XP84.C" "XP88.D" "XP30.N" "XP20.T" "XP54.T" "XP54.V"
##
## $NFV$BH
##  [1] "XP82.A" "XP84.A" "XP84.C" "XP88.D" "XP58.E" "XP10.F" "XP69.H"
##  [8] "XP33.M" "XP90.M" "XP30.N" "XP12.Q" "XP20.R" "XP20.T" "XP54.T"
## [15] "XP54.V" "XP76.V" "XP67.Y"
##
## $NFV$BY
## [1] "XP82.A" "XP84.A" "XP84.C" "XP88.D" "XP30.N" "XP20.R" "XP20.T" "XP54.T"
## [9] "XP54.V"
##
## $NFV$MarginalT
##  [1] "XP82.A" "XP84.A" "XP84.C" "XP88.D" "XP58.E" "XP10.F" "XP69.H"
##  [8] "XP33.M" "XP90.M" "XP30.N" "XP12.Q" "XP20.R" "XP20.T" "XP54.T"
## [15] "XP54.V" "XP76.V" "XP67.Y"
##
```

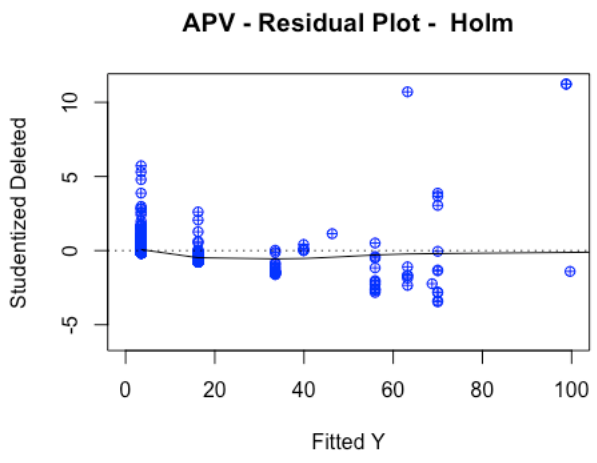
```
##
## $RTV
## $RTV$Holm
## [1] "XP54.A" "XP68.E" "XP33.F" "XP67.F" "XP24.I" "XP20.R" "XP47.V" "XP50.V"
## [9] "XP54.V"
##
## $RTV$Hochberg
## [1] "XP54.A" "XP68.E" "XP33.F" "XP67.F" "XP24.I" "XP20.R" "XP47.V" "XP50.V"
## [9] "XP54.V"
##
## $RTV$Hommel
## [1] "XP54.A" "XP68.E" "XP33.F" "XP67.F" "XP24.I" "XP20.R" "XP47.V" "XP50.V"
## [9] "XP54.V"
##
## $RTV$BonF
## [1] "XP54.A" "XP68.E" "XP33.F" "XP24.I" "XP20.R" "XP47.V" "XP50.V" "XP54.V"
##
## $RTV$BH
##  [1] "XP54.A" "XP84.A" "XP58.E" "XP68.E" "XP33.F" "XP67.F" "XP24.I"
##  [8] "XP33.I" "XP36.L" "XP90.M" "XP34.Q" "XP20.R" "XP20.T" "XP43.T"
## [15] "XP47.V" "XP50.V" "XP54.V"
##
## $RTV$BY
##  [1] "XP54.A" "XP84.A" "XP68.E" "XP33.F" "XP67.F" "XP24.I" "XP20.R"
##  [8] "XP47.V" "XP50.V" "XP54.V"
##
## $RTV$MarginalT
##  [1] "XP54.A" "XP84.A" "XP58.E" "XP68.E" "XP33.F" "XP67.F" "XP24.I"
##  [8] "XP33.I" "XP36.L" "XP90.M" "XP34.Q" "XP20.R" "XP20.T" "XP43.T"
## [15] "XP47.V" "XP50.V" "XP54.V"
##
##
## $SQV
## $SQV$Holm
##  [1] "XP73.A" "XP84.A" "XP84.C" "XP67.F" "XP48.M" "XP54.S" "XP54.T"
##  [8] "XP82.T" "XP48.V" "XP54.V" "XP84.V"
##
## $SQV$Hochberg
##  [1] "XP73.A" "XP84.A" "XP84.C" "XP67.F" "XP48.M" "XP54.S" "XP54.T"
##  [8] "XP82.T" "XP48.V" "XP54.V" "XP84.V"
##
## $SQV$Hommel
##  [1] "XP73.A" "XP84.A" "XP84.C" "XP67.F" "XP48.M" "XP54.S" "XP54.T"
##  [8] "XP82.T" "XP48.V" "XP54.V" "XP84.V"
##
## $SQV$BonF
##  [1] "XP73.A" "XP84.A" "XP84.C" "XP67.F" "XP48.M" "XP54.S" "XP54.T"
##  [8] "XP82.T" "XP48.V" "XP54.V" "XP84.V"
##
## $SQV$BH
##  [1] "XP22.A" "XP54.A" "XP73.A" "XP82.A" "XP84.A" "XP84.C" "XP83.D"
##  [8] "XP67.F" "XP53.L" "XP48.M" "XP54.S" "XP73.S" "XP91.S" "XP43.T"
## [15] "XP54.T" "XP82.T" "XP22.V" "XP48.V" "XP54.V" "XP84.V"
##
## $SQV$BY
##  [1] "XP73.A" "XP84.A" "XP84.C" "XP67.F" "XP48.M" "XP54.S" "XP54.T"
##  [8] "XP82.T" "XP22.V" "XP48.V" "XP54.V" "XP84.V"
##
## $SQV$MarginalT
##  [1] "XP22.A" "XP54.A" "XP73.A" "XP82.A" "XP84.A" "XP84.C" "XP83.D"
##  [8] "XP67.F" "XP53.L" "XP48.M" "XP54.S" "XP73.S" "XP91.S" "XP43.T"
## [15] "XP54.T" "XP82.T" "XP22.V" "XP48.V" "XP54.V" "XP84.V"

methods = list("Holm","Hochberg","Hommel", "Bonferroni","BH", "BY","MarginalT")
treatmenttypes = list("APV", "ATV","LPV","ITV","NFV","RTV","SQV")
```

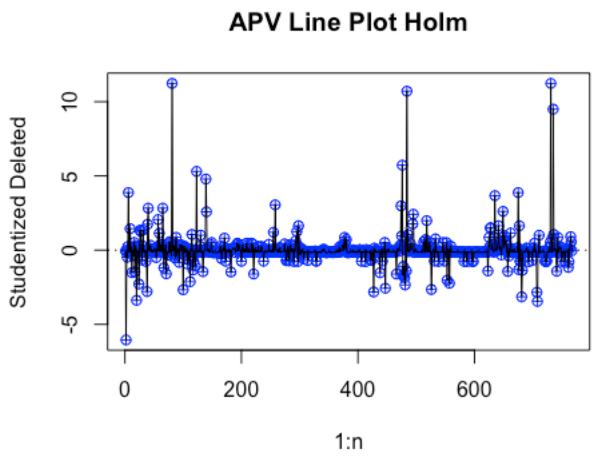
The output of this function will be a list of the top mutations as determined by each model to a certain confidence level. The full model will use all the different mutations (predictors) but have no intercept. After analysis we can see that one of the true mutation positions is taken out due to appearing less than 3 times. After we have run through the p-adjust module it will adjust the p-values output to create a false detection rate or family wise rate at a much lower level than the marginal T-test p-values. We then limit for each that all the p-values from the model are less than alpha. In further data exploration below, we will see these models through plots and R-squared. The posnum function is a function that will only get the numeric values. As we only need the position number not mutation type, this is used above and more in evaluation.

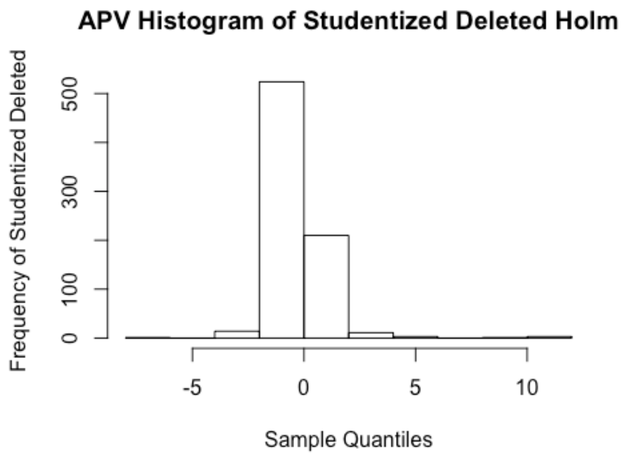
(Plots only shown for APV to not overwhelm, analyzed all of them as it is just pasting the loop and changing to dfiferent drug names) Exploratory data here:

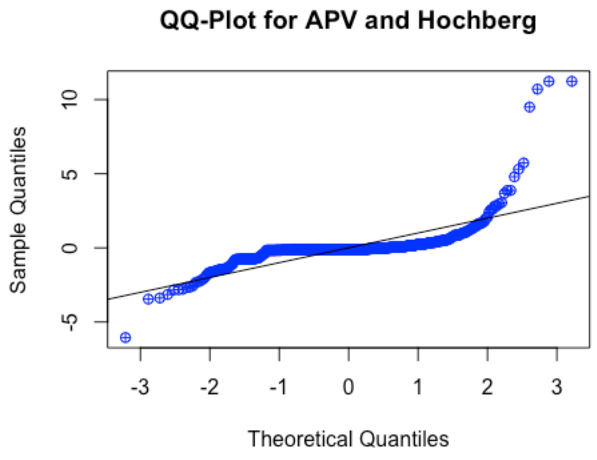
```
library(dplyr)

#APV
  missing = is.na(Y[1])
  YX = bind_cols(Y[1], data.frame(X))
  YX = YX[!missing,]
  YX = YX[,colSums(YX) >= 3]
  YX = YX[,colSums(abs(cor(YX)-1) < 1e-4) == 1]
  p = coef(summary(lm(APV~., data = YX)))[,4]
  #adjusted P-Values
  p.adjust.M <- p.adjust.methods[p.adjust.methods != "fdr"]
  p.adj    <- sapply(p.adjust.M, function(meth) p.adjust(p, meth))
  p.adj.2 <- sapply(p.adjust.M, function(meth) p.adjust(p, meth, n = length(p)))
  stopifnot(identical(p.adj[,"none"], p), p.adj <= p.adj.2)

  #Storing the different methods
  Holm = summary(lm(APV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,1]])))$r.squared
  Hochberg = summary(lm(APV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,2]])))$r.squared
  Hommel= summary(lm(APV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,3]])))$r.squared
  Bonferroni= summary(lm(APV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,4]])))$r.squared
  BH= summary(lm(APV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,5]])))$r.squared
  BY= summary(lm(APV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,6]])))$r.squared
  MarginalT = summary(lm(APV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,7]])))$r.squared
  APV = list(Holm = Holm, Hochberg = Hochberg, Hommel = Hommel,BonF =
Bonferroni,BH = BH,BY = BY, MarginalT = MarginalT)

  for(k in 1:7){
    mlm = (lm(APV~., data = bind_cols(YX[1], YX[(p.adj<alpha)[,k]])))
  qqnorm(rstudent(mlm), main= paste("QQ-Plot for", names(YX[1]), "and",
methods[k]),xlab="Theoretical Quantiles",ylab="Sample Quantiles",col =
"blue",pch=10)
    abline(a=0,b=1,lty=1,col="black")

    plot(predict(mlm),rstudent(mlm),main=paste(names(YX[1]), "- Residual Plot -
",methods[k]),xlab="Fitted Y",ylab="Studentized Deleted",col = "blue",pch=10,
xlim = c(0,100))
abline(h=0,lty=3)
lines(supsmu(predict(mlm),rstudent(mlm)),col="black")

n = length(bind_cols(YX[1], YX[(p.adj<alpha)[,k]])$APV)
plot(1:n,rstudent(mlm),main=paste(names(YX[1]),"Line
Plot",methods[k]),ylab="Studentized Deleted",col = "blue",pch=10)
abline(h=0,lty=3)
lines(1:n,rstudent(mlm),col="black",lty=1)

hist(rstudent(mlm),main=paste(names(YX[1]),"Histogram of Studentized
Deleted",methods[k]),ylab="Frequency of Studentized Deleted", xlab = "Sample
Quantiles")
}
```

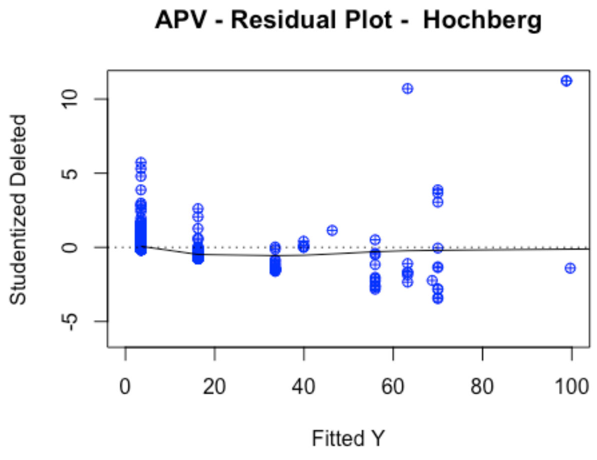

QQ-Plot for APV and Holm
Sample Quantiles
Theoretical Quantiles
-5
0
5
10
-3
-2
-1
0
1
2
3

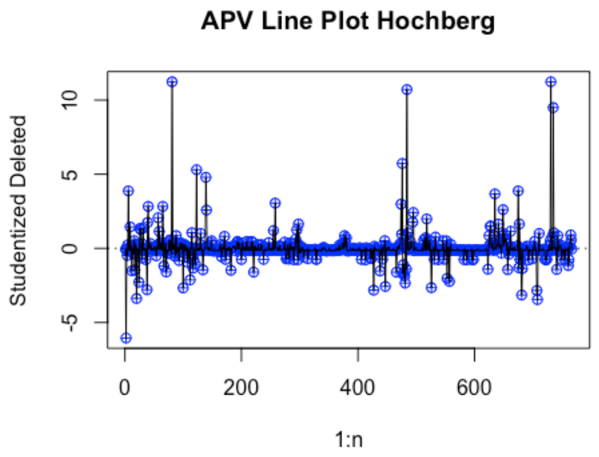

APV - Residual Plot - Holm
Studentized Deleted
Fitted Y
-5
0
5
10
0
20
40
60
80
100

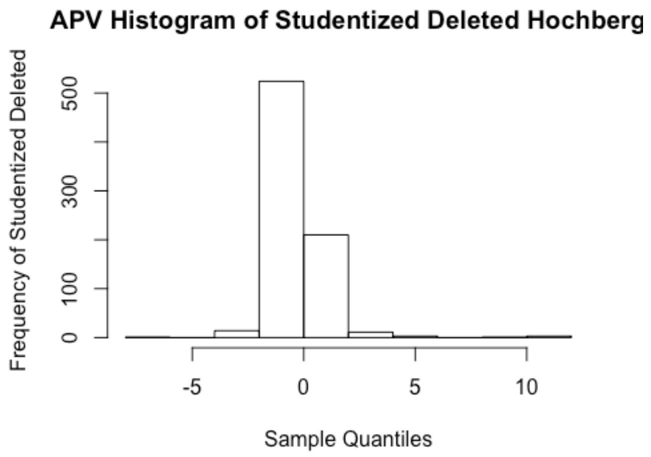

APV Line Plot Holm
Studentized Deleted
1:n
-5
0
5
10
0
200
400
600

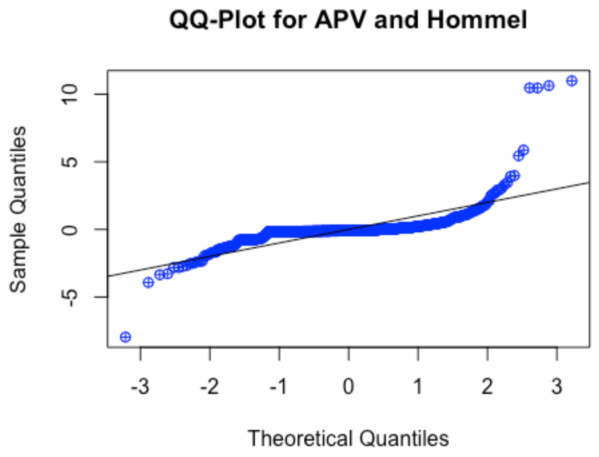

APV Histogram of Studentized Deleted Holm
Frequency of Studentized Deleted
Sample Quantiles
0
100
300
500
-5
0
5
10

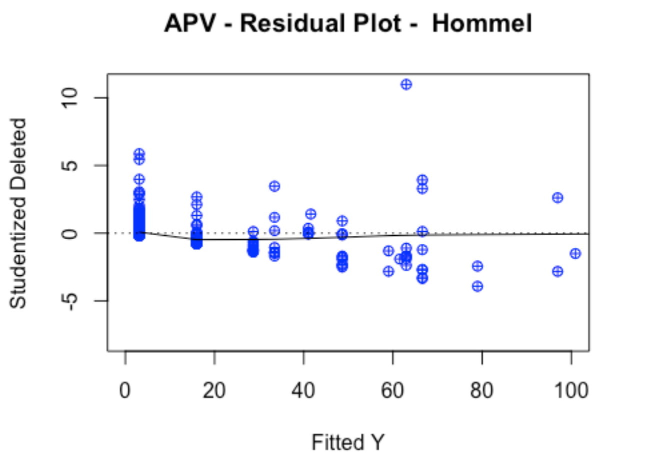

QQ-Plot for APV and Hochberg
Sample Quantiles
Theoretical Quantiles
-5
0
5
10
-3
-2
-1
0
1
2
3

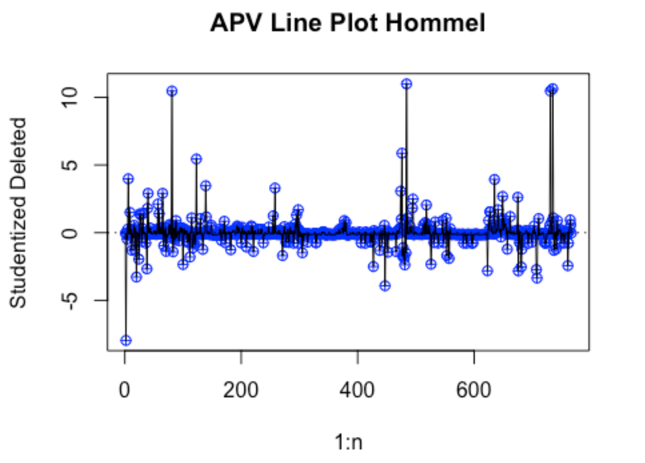

APV - Residual Plot - Hochberg
Studentized Deleted
Fitted Y
-5
0
5
10
0
20
40
60
80
100

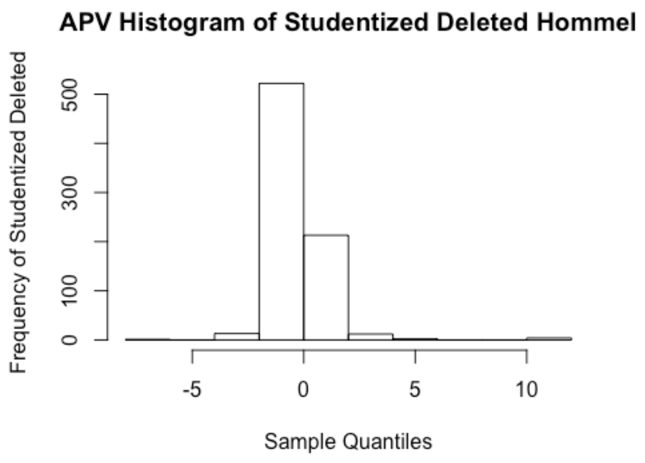

APV Line Plot Hochberg
Studentized Deleted
1:n
-5
0
5
10
0
200
400
600


APV Histogram of Studentized Deleted Hochberg
Frequency of Studentized Deleted
Sample Quantiles
0
100
300
500
-5
0
5
10

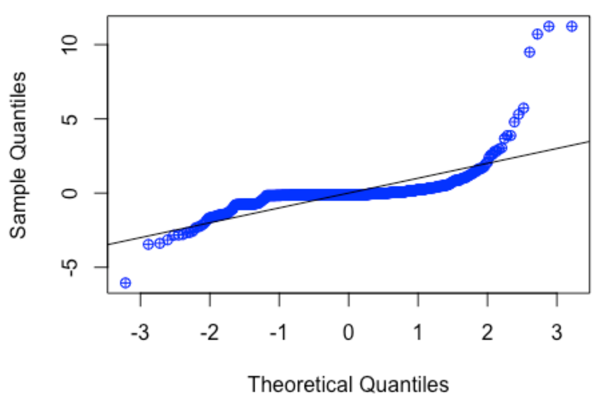

QQ-Plot for APV and Hommel
Sample Quantiles
Theoretical Quantiles
-5
0
5
10
-3
-2
-1
0
1
2
3


APV - Residual Plot - Hommel
Studentized Deleted
Fitted Y
10
5
0
-5
0
20
40
60
80
100

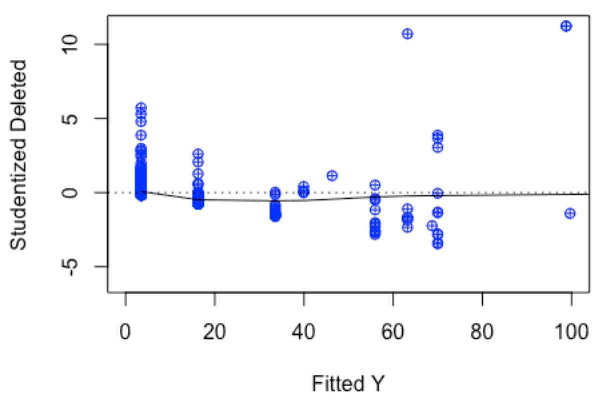

APV Line Plot Hommel
Studentized Deleted
1:n
10
5
0
-5
0
200
400
600


APV Histogram of Studentized Deleted Hommel
Frequency of Studentized Deleted
Sample Quantiles
500
300
100
0
-5
0
5
10

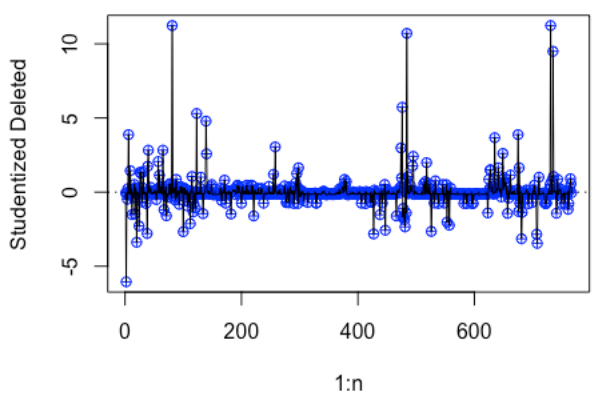
QQ-Plot for APV and Bonferroni

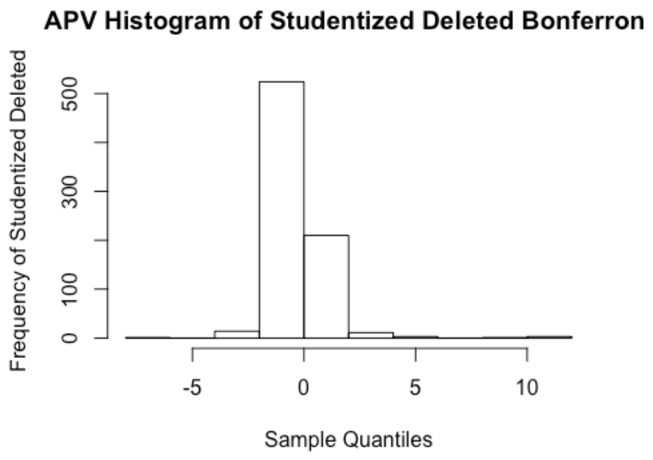


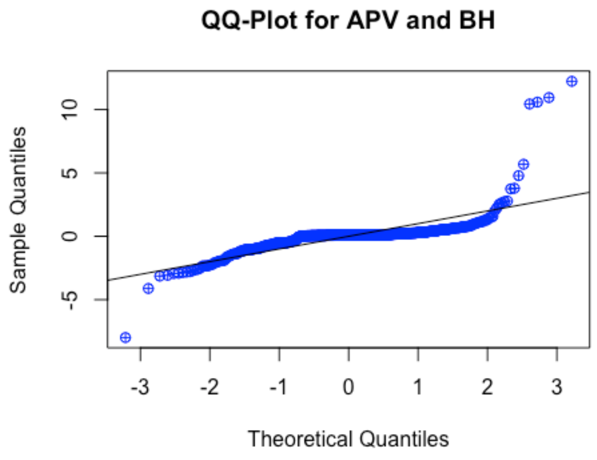
APV - Residual Plot - Bonferroni

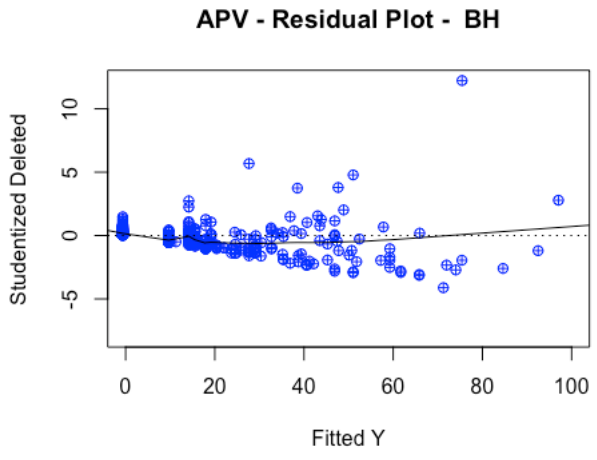


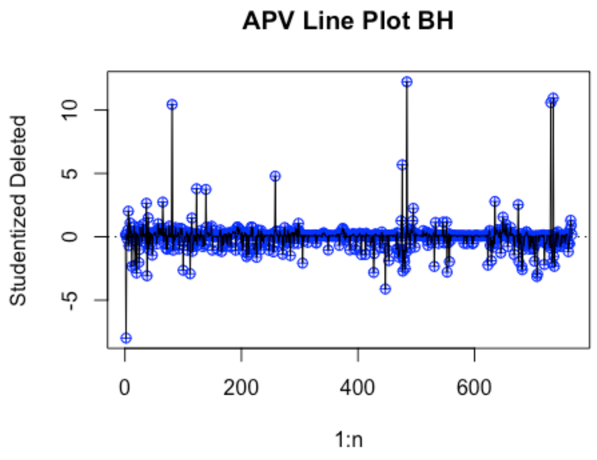
APV Line Plot Bonferroni

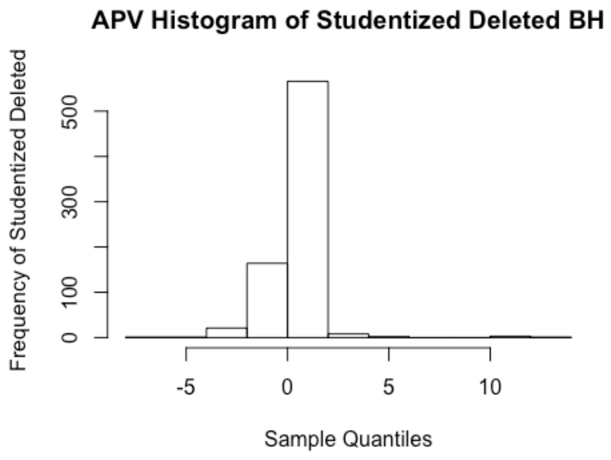


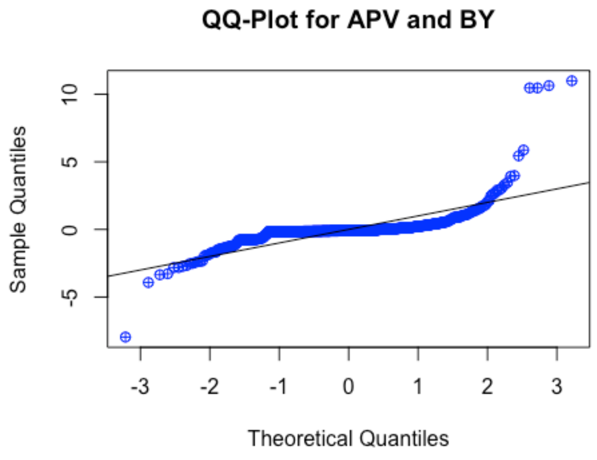

APV Histogram of Studentized Deleted Bonferron
Frequency of Studentized Deleted
0
100
300
500
-5
0
5
10
Sample Quantiles

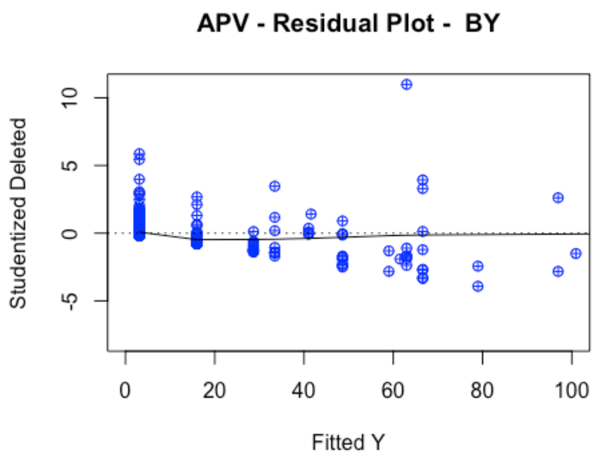

QQ-Plot for APV and BH
Sample Quantiles
-5
0
5
10
-3
-2
-1
0
1
2
3
Theoretical Quantiles


APV - Residual Plot - BH
Studentized Deleted
-5
0
5
10
0
20
40
60
80
100
Fitted Y

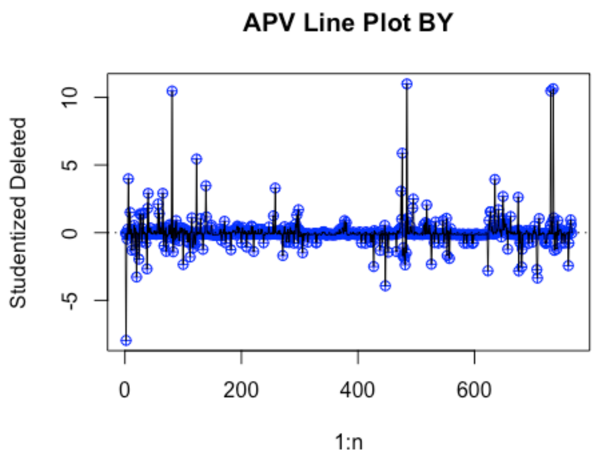

APV Line Plot BH
Studentized Deleted
-5
0
5
10
0
200
400
600
1:n


APV Histogram of Studentized Deleted BH
Frequency of Studentized Deleted
0
100
300
500
-5
0
5
10
Sample Quantiles

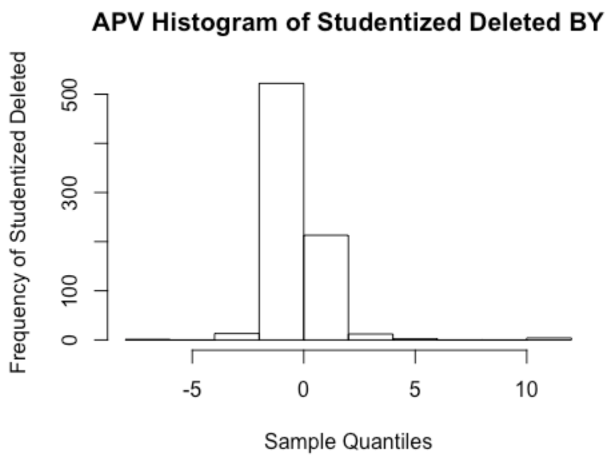

QQ-Plot for APV and BY
Sample Quantiles
-5
0
5
10
-3
-2
-1
0
1
2
3
Theoretical Quantiles



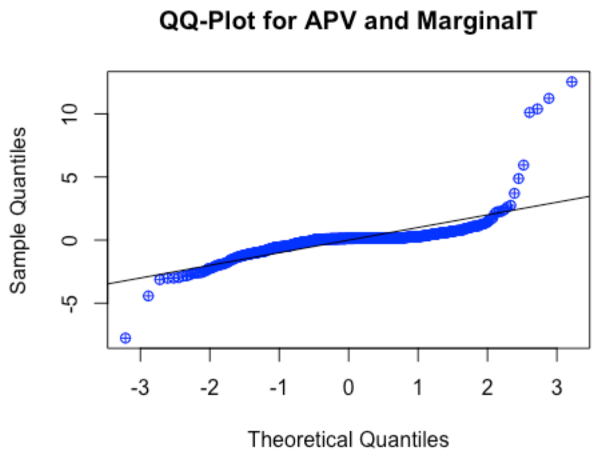
APV - Residual Plot - BY
Studentized Deleted
Fitted Y
-5
0
5
10
0
20
40
60
80
100

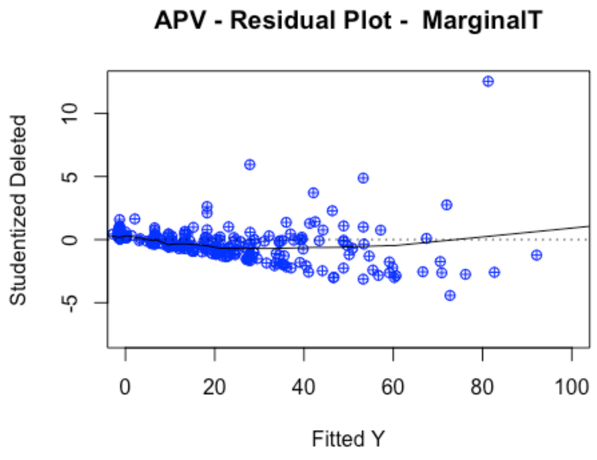


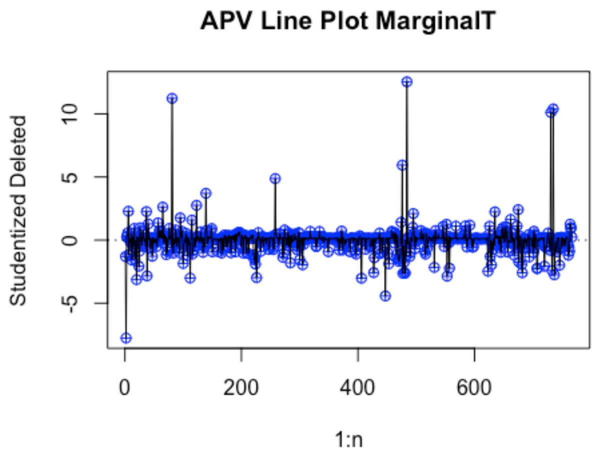
APV Line Plot BY
Studentized Deleted
1:n
-5
0
5
10
0
200
400
600

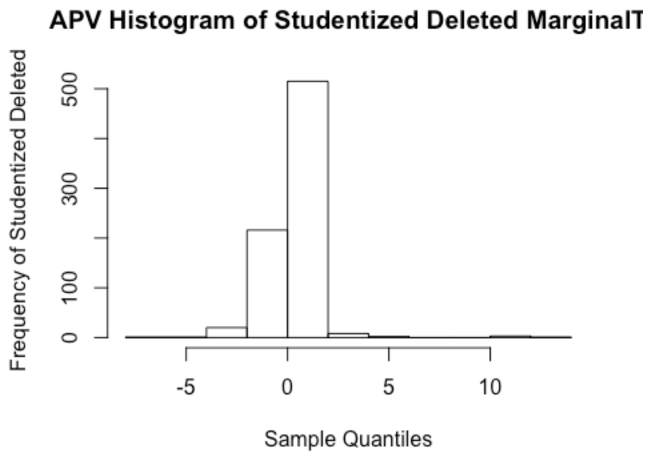


APV Histogram of Studentized Deleted BY
Frequency of Studentized Deleted
Sample Quantiles
0
100
300
500
-5
0
5
10


QQ-Plot for APV and MarginalT
Sample Quantiles
Theoretical Quantiles
-5
0
5
10
-3
-2
-1
0
1
2
3


APV - Residual Plot - MarginalT
Studentized Deleted
Fitted Y
-5
0
5
10
0
20
40
60
80
100





```
  #ATV
  missing = is.na(Y[2])
  YX = bind_cols(Y[2], data.frame(X))
  YX = YX[!missing,]
  YX = YX[,colSums(YX) >= 3]
  YX = YX[,colSums(abs(cor(YX)-1) < 1e-4) == 1]
  p = coef(summary(lm(ATV~., data = YX)))[,4]
  #adjusted P-Values
  p.adjust.M <- p.adjust.methods[p.adjust.methods != "fdr"]
  p.adj    <- sapply(p.adjust.M, function(meth) p.adjust(p, meth))
  p.adj.2 <- sapply(p.adjust.M, function(meth) p.adjust(p, meth, n = length(p)))
  stopifnot(identical(p.adj[,"none"], p), p.adj <= p.adj.2)

  #Storing the different methods
  Holm = summary(lm(ATV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,1]])))$r.squared
  Hochberg = summary(lm(ATV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,2]])))$r.squared
  Hommel= summary(lm(ATV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,3]])))$r.squared
```

```r
  Bonferroni= summary(lm(ATV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,4]])))$r.squared
  BH= summary(lm(ATV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,5]])))$r.squared
  BY= summary(lm(ATV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,6]])))$r.squared
  MarginalT = summary(lm(ATV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,7]])))$r.squared
  ATV = list(Holm = Holm, Hochberg = Hochberg, Hommel = Hommel,BonF =
Bonferroni,BH = BH,BY = BY, MarginalT = MarginalT)

  #IDV
  missing = is.na(Y[3])
  YX = bind_cols(Y[3], data.frame(X))
  YX = YX[!missing,]
  YX = YX[,colSums(YX) >= 3]
  YX = YX[,colSums(abs(cor(YX)-1) < 1e-4) == 1]
  p = coef(summary(lm(IDV~., data = YX)))[,4]
  #adjusted P-Values
  p.adjust.M <- p.adjust.methods[p.adjust.methods != "fdr"]
  p.adj    <- sapply(p.adjust.M, function(meth) p.adjust(p, meth))
  p.adj.2 <- sapply(p.adjust.M, function(meth) p.adjust(p, meth, n = length(p)))
  stopifnot(identical(p.adj[,"none"], p), p.adj <= p.adj.2)

  #Storing the different methods
  Holm = summary(lm(IDV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,1]])))$r.squared
  Hochberg = summary(lm(IDV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,2]])))$r.squared
  Hommel= summary(lm(IDV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,3]])))$r.squared
  Bonferroni= summary(lm(IDV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,4]])))$r.squared
  BH= summary(lm(IDV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,5]])))$r.squared
  BY= summary(lm(IDV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,6]])))$r.squared
  MarginalT = summary(lm(IDV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,7]])))$r.squared
  IDV = list(Holm = Holm, Hochberg = Hochberg, Hommel = Hommel,BonF =
Bonferroni,BH = BH,BY = BY, MarginalT = MarginalT)

  #LPV
  missing = is.na(Y[4])
  YX = bind_cols(Y[4], data.frame(X))
  YX = YX[!missing,]
  YX = YX[,colSums(YX) >= 3]
  YX = YX[,colSums(abs(cor(YX)-1) < 1e-4) == 1]
  p = coef(summary(lm(LPV~., data = YX)))[,4]
  #adjusted P-Values
  p.adjust.M <- p.adjust.methods[p.adjust.methods != "fdr"]
  p.adj    <- sapply(p.adjust.M, function(meth) p.adjust(p, meth))
  p.adj.2 <- sapply(p.adjust.M, function(meth) p.adjust(p, meth, n = length(p)))
  stopifnot(identical(p.adj[,"none"], p), p.adj <= p.adj.2)

  #Storing the different methods
  Holm = summary(lm(LPV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,1]])))$r.squared
  Hochberg = summary(lm(LPV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,2]])))$r.squared
  Hommel= summary(lm(LPV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,3]])))$r.squared
  Bonferroni= summary(lm(LPV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,4]])))$r.squared
  BH= summary(lm(LPV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,5]])))$r.squared
  BY= summary(lm(LPV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,6]])))$r.squared
  MarginalT = summary(lm(LPV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,7]])))$r.squared
  LPV = list(Holm = Holm, Hochberg = Hochberg, Hommel = Hommel,BonF =
Bonferroni,BH = BH,BY = BY, MarginalT = MarginalT)

  #NFV
  missing = is.na(Y[5])
  YX = bind_cols(Y[5], data.frame(X))
  YX = YX[!missing,]
  YX = YX[,colSums(YX) >= 3]
  YX = YX[,colSums(abs(cor(YX)-1) < 1e-4) == 1]
  p = coef(summary(lm(NFV~., data = YX)))[,4]
  #adjusted P-Values
  p.adjust.M <- p.adjust.methods[p.adjust.methods != "fdr"]
  p.adj    <- sapply(p.adjust.M, function(meth) p.adjust(p, meth))
  p.adj.2 <- sapply(p.adjust.M, function(meth) p.adjust(p, meth, n = length(p)))
  stopifnot(identical(p.adj[,"none"], p), p.adj <= p.adj.2)

  #Storing the different methods
  Holm = summary(lm(NFV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,1]])))$r.squared
  Hochberg = summary(lm(NFV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,2]])))$r.squared
  Hommel= summary(lm(NFV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,3]])))$r.squared
  Bonferroni= summary(lm(NFV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,4]])))$r.squared
  BH= summary(lm(NFV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,5]])))$r.squared
  BY= summary(lm(NFV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,6]])))$r.squared
  MarginalT = summary(lm(NFV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,7]])))$r.squared
  NFV = list(Holm = Holm, Hochberg = Hochberg, Hommel = Hommel,BonF =
Bonferroni,BH = BH,BY = BY, MarginalT = MarginalT)
```

```r
  #RTV
  missing = is.na(Y[6])
  YX = bind_cols(Y[6], data.frame(X))
  YX = YX[!missing,]
  YX = YX[,colSums(YX) >= 3]
  YX = YX[,colSums(abs(cor(YX)-1) < 1e-4) == 1]
  p = coef(summary(lm(RTV~., data = YX)))[,4]
  #adjusted P-Values
  p.adjust.M <- p.adjust.methods[p.adjust.methods != "fdr"]
  p.adj    <- sapply(p.adjust.M, function(meth) p.adjust(p, meth))
  p.adj.2 <- sapply(p.adjust.M, function(meth) p.adjust(p, meth, n = length(p)))
  stopifnot(identical(p.adj[,"none"], p), p.adj <= p.adj.2)

  #Storing the different methods
  Holm = summary(lm(RTV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,1]])))$r.squared
  Hochberg = summary(lm(RTV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,2]])))$r.squared
  Hommel= summary(lm(RTV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,3]])))$r.squared
  Bonferroni= summary(lm(RTV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,4]])))$r.squared
  BH= summary(lm(RTV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,5]])))$r.squared
  BY= summary(lm(RTV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,6]])))$r.squared
  MarginalT = summary(lm(RTV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,7]])))$r.squared
  RTV = list(Holm = Holm, Hochberg = Hochberg, Hommel = Hommel,BonF =
Bonferroni,BH = BH,BY = BY, MarginalT = MarginalT)

  #SQV
  missing = is.na(Y[7])
  YX = bind_cols(Y[7], data.frame(X))
  YX = YX[!missing,]
  YX = YX[,colSums(YX) >= 3]
  YX = YX[,colSums(abs(cor(YX)-1) < 1e-4) == 1]
  p = coef(summary(lm(SQV~., data = YX)))[,4]
  #adjusted P-Values
  p.adjust.M <- p.adjust.methods[p.adjust.methods != "fdr"]
  p.adj    <- sapply(p.adjust.M, function(meth) p.adjust(p, meth))
  p.adj.2 <- sapply(p.adjust.M, function(meth) p.adjust(p, meth, n = length(p)))
  stopifnot(identical(p.adj[,"none"], p), p.adj <= p.adj.2)

  #Storing the different methods

  Holm = summary(lm(SQV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,1]])))$r.squared
  Hochberg = summary(lm(SQV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,2]])))$r.squared
  Hommel= summary(lm(SQV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,3]])))$r.squared
```

```
  Bonferroni= summary(lm(SQV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,4]])))$r.squared
  BH= summary(lm(SQV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,5]])))$r.squared
  BY= summary(lm(SQV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,6]])))$r.squared
  MarginalT = summary(lm(SQV~., data = bind_cols(YX[1],
YX[(p.adj<alpha)[,7]])))$r.squared
  SQV = list(Holm = Holm, Hochberg = Hochberg, Hommel = Hommel,BonF =
Bonferroni,BH = BH,BY = BY, MarginalT = MarginalT)

  rsquared = list(APV = APV, ATV = ATV, IDV=IDV, LPV =LPV, NFV=NFV, RTV=RTV, SQV=
SQV)
```

R-squared for each drug and model produced

```
rsq <- data.frame(matrix(unlist(rsquared), nrow=7, byrow=T))
rownames(rsq) = treatmenttypes
colnames(rsq) = methods
rsq

##           Holm   Hochberg     Hommel Bonferroni        BH        BY
## APV 0.48315208 0.48315208 0.50534464 0.48315208 0.6325267 0.5053446
## ATV 0.05406145 0.05406145 0.05406145 0.05406145 0.3387323 0.0000000
## LPV 0.51837191 0.51837191 0.51837191 0.50142066 0.5748734 0.5354272
## ITV 0.40355539 0.40355539 0.40355539 0.40355539 0.6217229 0.4035554
## NFV 0.52286593 0.52286593 0.52286593 0.52286593 0.6450134 0.5228659
## RTV 0.52793419 0.52793419 0.52793419 0.52793419 0.6246624 0.5421290
## SQV 0.51929934 0.51929934 0.51929934 0.51929934 0.6087848 0.5333410
##     MarginalT
## APV 0.6680129
## ATV 0.5967156
## LPV 0.6694346
## ITV 0.7037542
## NFV 0.6911380
## RTV 0.7123601
## SQV 0.6445199
```

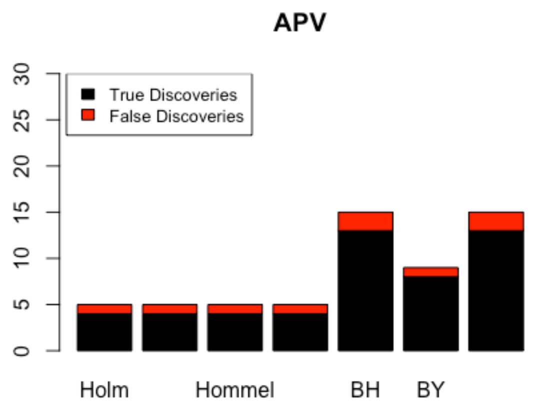
Evaluating the results

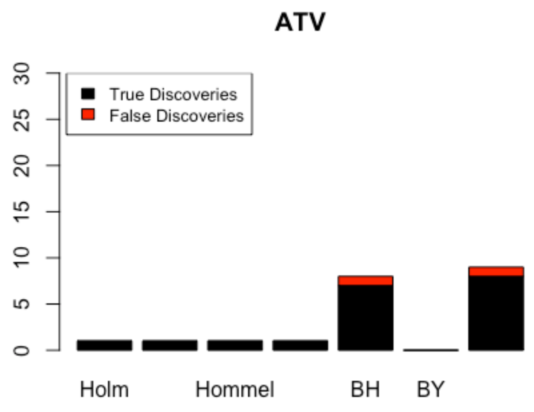
In this case, we are fortunate enough to have a "ground truth" obtained by another experiment (data saved as `tsm_df`). Using this, we can evaluate the selected results. Note that we only need to compare the position of the mutations, not the mutation type. This is because it is known that multiple mutations at the same protease or RT position can often be associated with related drug-resistance outcomes.

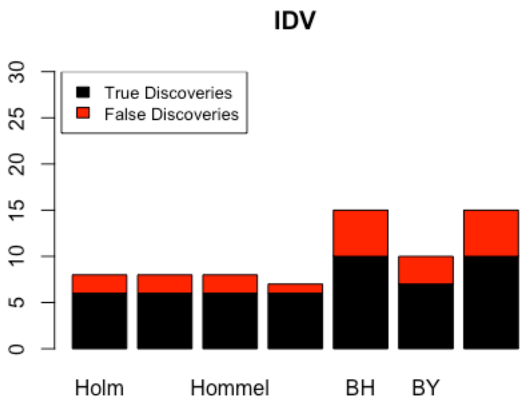
```
rates <- lapply(results, function(treatment) {
  lapply(treatment, function(selected) {
    positions = unique(posnum(selected))
    Falsem = setdiff(positions, tsm_df$Position)
    Missed = setdiff(tsm_df$Position, positions)
    Truem = setdiff(positions, Falsem)
    list(Truem = length(Truem), Falsem = length(Falsem), FDR = length(Falsem) /
length(positions), Missed = length(Missed))
  })
})
```

Here we will use a bar graph to depict the false detection and true detection amount for each drug treatment used. We can see that the amount of variables for each model varies significantly under different drugs.

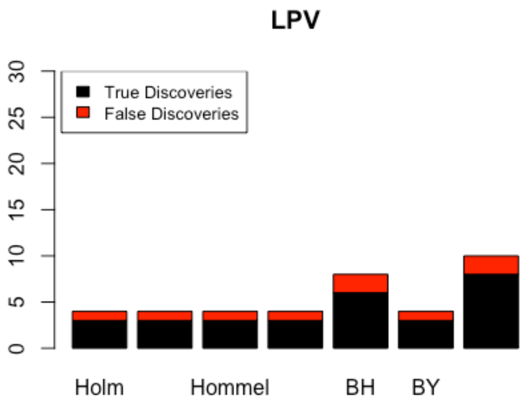
```
for (treatment in names(rates)) {
  pdata = do.call(cbind, rates[[treatment]])[c('Truem','Falsem'),]
  barplot(pdata, main = paste(treatment), col = c('black','red'), ylim = c(0,30))
  legend(0,30,legend=c("True Discoveries","False Discoveries"),
fill=c('black','red'), cex=0.8)
}
```



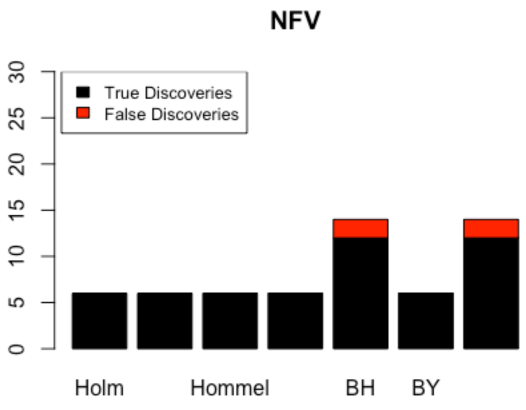




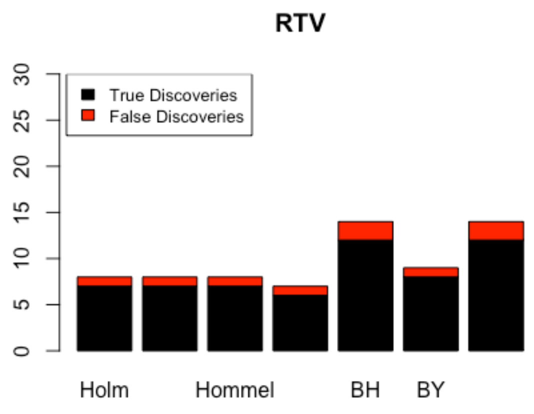
IDV
True Discoveries
False Discoveries
0
5
10
15
20
25
30
Holm
Hommel
BH
BY

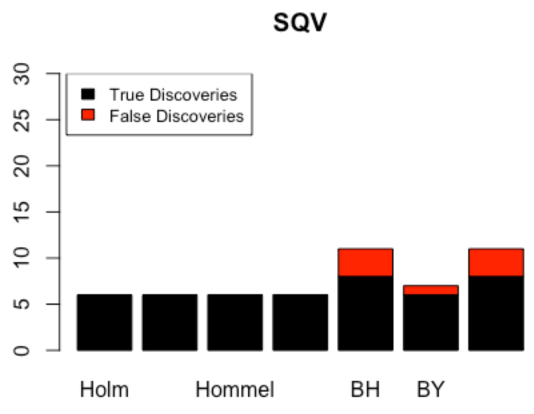


LPV
True Discoveries
False Discoveries
0
5
10
15
20
25
30
Holm
Hommel
BH
BY



NFV
True Discoveries
False Discoveries
0
5
10
15
20
25
30
Holm
Hommel
BH
BY





Below, we will print out a list of all True mutation positions found, False mutation positions, and Missed mutation positions for each model we used. This will then be put again through to determine the best statistical model

```r
percenttotal <- function(results){
  rates1 <- lapply(results, function(treatment) {
  lapply(treatment, function(selected) {
    positions = unique(posnum(selected))
    positions
})})
rates1
}

totalpos<- function(rates1){
Holm = sort(unique(unname(unlist(sapply(rates1, function(x) x[[1]])))))
Hochberg = sort(unique(unname(unlist(sapply(rates1, function(x) x[[2]])))))
Hommel = sort(unique(unname(unlist(sapply(rates1, function(x) x[[3]])))))
Bonferroni = sort(unique(unname(unlist(lapply(rates1, function(x) x[[4]])))))
BH = sort(unique(unname(unlist(sapply(rates1, function(x) x[[5]])))))
BY = sort(unique(unname(unlist(lapply(rates1, function(x) x[[6]])))))
Marginal = sort(unique(unname(unlist(sapply(rates1, function(x) x[[7]])))))
list2 = list(Holm = Holm, Hochberg = Hochberg, Hommel = Hommel,BonF =
Bonferroni,BH = BH,BY = BY, MarginalT = Marginal)
list2}

perc <- function(list2){
Missed = setdiff(tsm_df$Position, list2)
Falsem = setdiff(list2, tsm_df$Position)
True = setdiff(list2, Falsem)
list(Truem = True, Missed = Missed, Falsem = Falsem)
}

PositionOutput = lapply(totalpos(percenttotal(results)), function(l) perc(l))
PositionOutput

## $Holm
## $Holm$Truem
##  [1] 11 20 24 30 32 33 43 47 48 50 54 67 71 73 76 82 84 88
##
## $Holm$Missed
##  [1] 10 23 34 35 46 53 55 58 66 74 79 85 89 90 92 95
##
## $Holm$Falsem
## [1] 57 60 68 69
##
##
## $Hochberg
## $Hochberg$Truem
##  [1] 11 20 24 30 32 33 43 47 48 50 54 67 71 73 76 82 84 88
##
## $Hochberg$Missed
##  [1] 10 23 34 35 46 53 55 58 66 74 79 85 89 90 92 95
##
## $Hochberg$Falsem
## [1] 57 60 68 69
##
##
## $Hommel
## $Hommel$Truem
##  [1] 11 20 24 30 32 33 43 47 48 50 54 67 71 73 76 82 84 88
##
## $Hommel$Missed
##  [1] 10 23 34 35 46 53 55 58 66 74 79 85 89 90 92 95
##
## $Hommel$Falsem
## [1] 57 60 68 69
##
##
## $BonF
## $BonF$Truem
##  [1] 11 20 24 30 32 33 43 47 48 50 54 67 71 73 76 82 84 88
##
```

```
## $BonF$Missed
##  [1] 10 23 34 35 46 53 55 58 66 74 79 85 89 90 92 95
##
## $BonF$Falsem
## [1] 60 68 69
##
##
## $BH
## $BH$Truem
##  [1] 10 11 20 24 30 32 33 34 43 46 47 48 50 53 54 55 58 67 71 73 74 76 82
## [24] 84 85 88 89 90 92
##
## $BH$Missed
## [1] 23 35 66 79 95
##
## $BH$Falsem
##  [1] 12 19 22 36 57 60 68 69 70 72 83 91
##
##
## $BY
## $BY$Truem
##  [1] 11 20 24 30 32 33 34 43 47 48 50 54 67 73 76 82 84 88 89
##
## $BY$Missed
##  [1] 10 23 35 46 53 55 58 66 71 74 79 85 90 92 95
##
## $BY$Falsem
## [1] 12 22 57 60 68 69
##
##
## $MarginalT
## $MarginalT$Truem
##  [1] 10 11 20 24 30 32 33 34 43 46 47 48 50 53 54 55 58 67 71 73 74 76 82
## [24] 84 85 88 89 90 92
##
## $MarginalT$Missed
## [1] 23 35 66 79 95
##
## $MarginalT$Falsem
##  [1] 12 19 22 36 57 60 68 69 70 72 83 91
```

```r
Evaluation = function(method){
  True = method$True
  Missed = method$Missed
  False = method$Falsem
  length(True)/(length(Missed)+length(True))-
length(False)/(length(True)+length(False))
}
data2 = lapply(PositionOutput, function(y) Evaluation(y))
data2 <- data.frame(matrix(unlist(data2), nrow=1, byrow=T))
rownames(data2) = "Evaluation Final"
```

```
colnames(data2) = methods
data2

##                        Holm  Hochberg    Hommel Bonferroni        BH
## Evaluation Final 0.3475936 0.3475936 0.3475936  0.3865546 0.5602582
##                          BY MarginalT
## Evaluation Final 0.3188235 0.5602582
```

Evaluation and Conclusion

We can see that the best performing method under an alpha of 0.05 (0.005), happens to be the Benjamini-Hochberg False Detection Rate method. We get an optimal value for our test statistic (% of total discovered - % False Discovery) of 0.5602582. From the plots we can see a lot of variability and changing our statistical description could make way to another method being chosen as best. The graphs and the final also show there is significant overlap in the mutations chosen with each drug, making some better with more limited data, while others better when the data available is expansive. To note, the MarginalT alpha was taken at a factor of 10 to make it more in line with the other methods. Still the FDR with BH, (or the MarginalT under alpha of 0.005) proves to be the best under our statistical description. The adjustments to Bonferroni all appeared to lowered the the evaluation result, but it is worthy to note they all produced the same evaluation result, showing that maybe the more fine details of these methods are not statistically significant on the data set we are looking at. It is all worth noting that the the adjustments to bonferroni are better than Bonferroni when alpha becomes higher, as in extension. Since we are limiting to alpha of 0.05 though, they do not perform better..

Appendix and Further studies:

For further analysis we can use this method to determine the optimal alpha or FDR to also optimize the results. This is an example method to determine the optimal for each model

```
#Function to show the lists and max values, combine the different drugs into one
posnum <- function(x)
  sapply(regmatches(x, regexpr('[[:digit:]]+', x)), as.numeric)

x = c(1:40)

for (i in 1:40){
alpha = (i+1)/200
fdr = (i+1)/200
results = lapply(Y, function(y) selection(X,y,fdr, alpha))
alltypes = unique(c(results$APV$BH, results$ATV$BH, results$IDV$BH,
results$LPV$BH, results$NFV$BH, results$RTV$BH, results$SQV$BH))
positions = unique(posnum(alltypes))
Missed = setdiff(tsm_df$Position, positions)
Falsem = setdiff(positions, tsm_df$Position)
Truem = setdiff(positions, Falsem)
m = length(Truem)/(length(Missed)+length(Truem)) -
length(Falsem)/length(positions)
x[i] = m
}
cat("Combination max:", max(x))
```

```
## Combination max: 0.6727159
```

```
cat("\nalpha:",(match(max(x),x)+1)/200)
```

```
##
## alpha: 0.15
```

References

Chen, S. Y., Feng, Z., & Yi, X. (2017). A general introduction to adjustment for multiple comparisons. Journal of thoracic disease, 9(6), 1725–1729. doi:10.21037/jtd.2017.05.34

Haynes W. (2013) Holm's Method. In: Dubitzky W., Wolkenhauer O., Cho KH., Yokota H. (eds) Encyclopedia of Systems Biology. Springer, New York, NY

Rhee, Soo-Yon, W Jeffrey Fessel, Andrew R Zolopa, Leo Hurley, Tommy Liu, Jonathan Taylor, Dong Phuong Nguyen, et al. 2005. "HIV-1 Protease and Reverse-Transcriptase Mutations: correlations with Antiretroviral Therapy in Subtype B Isolates and Implications for Drug-Resistance Surveillance." *Journal of Infectious Diseases 192 (3). Oxford University Press: 456–65*.

Rhee, Soo-Yon, Jonathan Taylor, Gauhar Wadhera, Asa Ben-Hur, Douglas L Brutlag, and Robert W Shafer. 2006. "Genotypic Predictors of Human Immunodeficiency Virus Type 1 Drug Resistance." *Proceedings of the National Academy of Sciences 103 (46). National Academy of Sciences: 17355–60*.