Report 1

Multiple Regression and Multiple Testing

Klaudia Balcer

12/17/2021

Contents

Introduction

Linear regression is a basic statistical method. It can model linear dependencies well. However, it is also prone to noise. Including to many variables in the model may cause a bad reality mapping. A simple way to select variables which should be included in the model is multiple testing. In this report we will show a simulation study on the efficiency of this approach. The structure of the document is as folllows. In the first section we will introduce the subject of multiple regression. After that we will focus on multiple testing in the second section. The third sections contains the theoretical calculations and results obtained from the simulation study.

Multiple Regression

Multiple Linear Regression is a statistical approach for modeling a linear relationship between a scalar response variable and many explanatory variables. Let's consider n observations (values of response variable and vectors of explanatory variables). The number of explanatory variables is denoted by $p-1$. The model is of the form:

$$Y_i = \beta_0 + X_i\beta_{[1,\dots,p-1]} + \epsilon_i$$

or equivalently the model in a matrix form:

$$Y = X\beta + \epsilon$$

where:

- Y_i is the value of the response variable for ith observation, Y is the (horizontal) vector of response variable $(n \times 1)$;
- X_i is the (vertical) vector of explanatory variables for ith observation, X is a design matrix (first column filled with ones, $X_{i,j+1}$ is the value of the jth explanatory variable in the ith observation $(n \times p)$;
- β_0 is the intercept,
- β is the (horizontal) vector of coefficients $(n \times 1)$,
- ϵ_i is the error for ith observation, let's consider normally distributed errors: $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, ϵ is the vector of errors $(n \times 1)$,
- i is the index of the observation.

There are two standard formulas for estimating β: least squares and maximum likelihood. When the error is normally distributed (as we have previously assumed) those method are equivalent.

The estimator of β is denoted by b, the estimator of σ (standard deviation of the errors) will be denoted as s.

The fitted values of the response variables will be denoted as $\hat{Y}$. We can calculate it by replacing β with its estimator and ϵ by its mean value (b and 0).

$$\hat{Y} = Xb$$

The difference between real and fitted (predicted) values of the response variable is called residual and it is marked with e:

$$e = Y - \hat{Y}$$

Let's get into more details.

Estimating Regression Coefficients

We will show the least squares estimation of β. The method is based on minimalizing the values of the error ϵ.

$$b = arg \min_{b \in \mathbb{R}^n} \quad (Y - Xb)^T (Y - Xb)$$

Let's expand the minimised formula:

$$(Y - Xb)^T (Y - Xb) = Y^T Y - b^T X^T Y - Y^T Xb + b^T X^T Xb$$

Let's derive by b:

$$-X^T Y - Y^T X + 2X^T Xb = 0$$

$$X^T Xb = X^T Y$$

The sationar point:

$$b = (X^T X)^{-1} X^T Y$$

The second derivative is $2X^T X$. Let's have $v \in \mathbb{R}^p$.

$$v^T X^T Xv = (Xv)^T Xv = Xv \circ Xv = \|Xv\|^2 \geq 0$$

Thus, $X^T X$ is a positive-definite matrix, so we have found the local minimum.

Let take a look on the distribution of the estimator:

$$b \sim \mathcal{N}\big(\beta, \sigma^2 (X^T X)^{-1}\big)$$

Fitted Values

After providing the formula for b we are able to derive the formula for $\hat{Y}$ in terms of X and Y:

$$\hat{Y} = Xb = X(X^T X)^{-1} X^T Y$$

As we can see in the above equation, there is a linear dependence between Y and $\hat{Y}$. The matrix transforming Y into $\hat{Y}$ is called hat matrix and it is denoted by H:

$$\hat{Y} = HY$$

The formual for the hat matrix is:

$$H = X(X^T X)^{-1} X^T$$

THe matrix H *projects* Y into $\hat{Y}$.

Estimator of σ

The estimator of σ^2 uses the values of the residuals has the form:

$$s^2 = \frac{e^T e}{n-p} = \frac{(Y - Xb)^T (Y - Xb)}{n-p}$$

Wishart and Inverse-Wishart Distribution

In the upcoming simulations, we will consider the design matrix of i.i.d. normally distributed random variables. X is a $n \times p$ matrix, each row is independently drawn from p-variate normal distribution with mean zero and covariance matrix $\Sigma = \frac{1}{1000} I$. Thus, $X^T X$ ($p \times p$ random matrix) has the Wishart probability distribution with parameters Σ and n:

$$X^T X \sim W_p(\Sigma, n)$$

Thus, the inverse of $X^T X$ comes from the Inverse-Wishart distribution:

$$(X^T X)^{-1} \sim W_p^{-1}(\Sigma^{-1}, n)$$

The mean value of a random variable from Inverse-Wishart distribution has the mean:

$$\mathbb{E}\big[(X^T X)^{-1}\big] = \frac{\Sigma^{-1}}{n-p-1}$$

Multiple Testing

Let's consider p testing problems:

$$H_{0,i} : \mu_i = 0 \quad vs \quad H_{1,i} : \mu_i \neq 0$$

Multiple testing means testing many individual hypotheses ($H_{0,i}$ vs. $H_{1,i}$) at the same time. To simply statistically test each hypothesis separately doesn't lead to satisfying results. We use **multiple comparison procedures** (MCPs) instead. MCPs are used to improve the quality of the tests.

The outcome of an MCP can be presented in the following form:

	accepted	rejected	total
true	TN (U)	FP (S)	p_0
false	FN (V)	TP (T)	$p - p_0$
total	p - R	R	p

The symbols used:

- TN - True Negatives - the null hypothesis is true, and it was accepted (U in *Candes*),
- FP - False Positives - the null hypothesis is true, but it was rejected (V in *Candes*),
- FN - False Negatives - the null hypothesis is false, and it was accepted (T in *Candes*),
- TP - True Positives - the null hypothesis is false, and it was rejected (S in *Candes*),
- p_0 - the number of true null hypotheses,

- R - the number of rejected hypotheses,
- p - the number of hypotheses.

Test Quality Measures

The definition of the Type I Error does not simply propagate to multiple testing problem. On the one hand, allowing a single false discovery with the probability α is a very strict condition. On the other hand, allowing the probability of false discovery in each test to be α, leads to many false discoveries. Hence we need to introduce new quality measures: FWER and FDR.

FWER

FWER stands for **F**amili**w**ise **E**rror **R**ate.

In **strong** sense: it is the probability of making any false discoveries:

$$FWER = \mathbb{P}(V \geq 1)$$

In **weak** sense: it is the probability of making any false discoveries if all the global null hypothesis is true:

$$FWER = \mathbb{P}(V \geq 1 | \forall_i H_{0,i})$$

FDR

FDR stands for **F**alse **D**iscovery **R**ate id the expected value of FDP (**F**alse **D**iscovery **P**roportion) - the ratio between the numbers of false discoveries and all rejections:

$$FDR = \mathbb{E}\left[\frac{FP}{max(R,1)}\right]$$

Under the global null hypothesis, FDR and FWEAR are equivalent.

mFDR

$$mFDR = \frac{\mathbb{E}V}{\mathbb{E}R} = \frac{\mathbb{E}V}{\mathbb{E}[V+T]}$$

Power

Power is the probability of rejecting the null hypothesis when it is false. In the above terms, we can express it as the expected value of the ratio of TP and the number of false hypotheses:

$$power = \mathbb{E}\left[\frac{TP}{p - p_0}\right]$$

Multiple Testing Procedures

For all below procedures, first we calculate the p-values of single tests:

- p-values: $p_1, p_2, \ldots, p_p$,
- ordered p-values: $p_{(1)}, p_{(2)}, \ldots, p_{(p)}$.

Bonferroni's procedure

Reject $H_{0,i}$ if:

$$p_i < \frac{\alpha}{n}$$

This method is very conservative. We know from the lecture that Bonferroni's method controls FWER in a strong sense. In fact,

$$FWER = \mathbb{P}(FP \geq 1) \leq \sum_{i=1}^{p_0} \mathbb{P}(FP = i) = \sum_{i=1}^{n_0} \binom{p_0}{i} \alpha^i (1-\alpha)^{p_0 - i}$$

$$FWER \leq \frac{p_0}{p}\alpha$$

Benjamini-Hochberg's procedure

Reject $H_{0,(i)}$ if:

$$\exists_{(j \geq i)} \quad p_{(j)} < \frac{j}{n}\alpha$$

Benjamini-Hochberg's method is a step-down procedure. This procedure controls FDR under independence. Thus, it controls FWER weakly. It does not control FWER in a strong sense. It is much more liberal than Hochberg's procedure (more powerful and leads to more false discoveries).

Simualtions

Theoretical calculations

Variance of the Estimators of Individual Regression Coefficients

For a fixed matrix of observation X, the variance of the estimators of individual regression coefficients is given by the formula:

$$\sigma^2_{\beta_i} = \sigma^2 (X^T X)^{-1}_{[i,i]}$$

When resampling the matrix of observations X, we replace $(X^T X)^{-1}_{[i,i]}$ by its expectation:

$$\mathbb{E}_X \sigma^2_{\beta_i} = \sigma^2 \mathbb{E}\Big[(X^T X)^{-1}_{[i,i]}\Big]$$

We know that $(X^T X)^{-1}$ has the inverse-Wishart distribution. Its expected value is given by the formula:

$$\mathbb{E}\big[(X^T X)^{-1}\big] = \frac{n \mathbb{I}_{n \times n}}{n - p - 1}$$

Thus:

$$\mathbb{E}_X \sigma^2_{\beta_i} = \sigma^2 \mathbb{E}\Big[(X^T X)^{-1}_{[i,i]}\Big] = \sigma^2 \frac{n}{n - p - 1}$$

1. n = 10

$$\mathbb{E}_X \sigma^2_{\beta_i} = \sigma^2 \frac{n}{n - p - 1} = 1 \cdot \frac{1000}{1000 - 9} = \frac{1000}{991} \approx 1.01$$

2. n = 100

$$\mathbb{E}_X \sigma^2_{\beta_i} = \sigma^2 \frac{n}{n-p-1} = 1 \cdot \frac{1000}{1000-99} = \frac{1000}{901} \approx 1.11$$

3. n = 500

$$\mathbb{E}_X \sigma^2_{\beta_i} = \sigma^2 \frac{n}{n-p-1} = 1 \cdot \frac{1000}{1000-499} = \frac{1000}{501} \approx 2.00$$

4. n = 950

$$\mathbb{E}_X \sigma^2_{\beta_i} = \sigma^2 \frac{n}{n-p-1} = 1 \cdot \frac{1000}{1000-949} = \frac{1000}{51} \approx 19.61$$

We expect the variance of the estimators to be very large for p close to n. Thus, the results of the experiment are expected to be not satisfying for p close to n.

FWER

$$FWER = \mathbb{P}(V > 0) = \mathbb{P}(FP > 0) = \mathbb{P}\Big(\bigcup_{i \in \mathcal{H}_0} V_i \Big)$$

When having fixed observation matrix X it is hard to provide the exact formula for FWER. However, we can easily derive an upper bound:

$$FWER = \mathbb{P}\Big(\bigcup_{i \in \mathcal{H}_0} V_i \Big) \leq \sum_{i \in \mathcal{H}_0} \mathbb{P}(V_i) = p_0 \cdot \alpha$$

When resampling the observations matrix, the individuial regression coeficients become independent:

$$\mathbb{E}_X \hat{\beta} \sim \mathcal{N}\big(\beta, \sigma^2 \mathbb{E}(X^T X^{-1})\big) = \mathcal{N}\big(\beta, \sigma^2 \frac{n\mathbb{I}}{n-p-1}\big)$$

Whis leads us to a formuka for FWER:

$$\mathbb{E}_X FWER = \mathbb{P}\Big(\bigcup_{i \in \mathcal{H}_0} \mathbb{E}_X V_i \Big) = 1 - \mathbb{P}\Big(\bigcap_{i \in \mathcal{H}_0} \mathbb{E}_X V_i \Big) = 1 - (1-\alpha)^{p_0}$$

n = 10

$$\mathbb{E}_X FWER = 1 - (1-\alpha)^{p_0} = 1 - (1-\alpha)^{p_0} \approx 0.41$$

n = 100

$$\mathbb{E}_X FWER = 1 - (1-\alpha)^{p_0} = 1 - (1-\alpha)^{p_0} \approx 0.41$$

n = 500

$$\mathbb{E}_X FWER = 1 - (1-\alpha)^{p_0} = 1 - (1-\alpha)^{p_0} \approx 0.41$$

n = 950

$$\mathbb{E}_X FWER = 1 - (1-\alpha)^{p_0} = 1 - (1-\alpha)^{p_0} \approx 0.41$$

The construction of Bonferroni's procedure changes the rejection threshold. Whus, we can derive the formula for FWER in a similar way:

$$FWER = \mathbb{P}\Big(\bigcup_{i\in\mathcal{H}_0} V_i\Big) \leq \sum_{i\in\mathcal{H}_0} \mathbb{P}(V_i) = p_0 \cdot \frac{\alpha}{n}$$

$$\mathbb{E}_X FWER = \mathbb{P}\Big(\bigcup_{i\in\mathcal{H}_0} \mathbb{E}_X V_i\Big) = 1 - \mathbb{P}\Big(\bigcap_{i\in\mathcal{H}_0} \mathbb{E}_X V_i\Big) = 1 - \prod_{i\in\mathcal{H}_0} \mathbb{P}\big(\mathbb{E}_X V_i\big) = 1 - \Big(1 - \frac{\alpha}{n}\Big)^{p_0}$$

n = 10

$$\mathbb{E}_X FWER = 1 - \Big(1 - \frac{\alpha}{n}\Big)^{p_0} = 1 - \Big(1 - \frac{0.1}{10}\Big)^{5} = 1 - 0.99^5 \approx 0.05$$

n = 100

$$\mathbb{E}_X FWER = 1 - \Big(1 - \frac{\alpha}{n}\Big)^{p_0} = 1 - \Big(1 - \frac{0.1}{100}\Big)^{5} = 1 - 0.999^5 \approx 0$$

n = 500

$$\mathbb{E}_X FWER = 1 - \Big(1 - \frac{\alpha}{n}\Big)^{p_0} = 1 - \Big(1 - \frac{0.1}{500}\Big)^{5} = 1 - 0.9998^5 \approx 0$$

n = 950

$$\mathbb{E}_X FWER = 1 - \Big(1 - \frac{\alpha}{n}\Big)^{p_0} = 1 - \Big(1 - \frac{0.1}{950}\Big)^{5} = 1 - 0.9998947^5 \approx 0$$

For Benjamini-Hochberg:

$$FWER = \mathbb{P}\Big(\bigcup_{i\in\mathcal{H}_0} V_i\Big) \leq \sum_{i\in\mathcal{H}_0} \mathbb{P}(V_i) = \sum_{i=6}^{p} \mathbb{P}(V_i)$$

$$\mathbb{E}_X FWER = \mathbb{P}\Big(\bigcup_{i\in\mathcal{H}_0} \mathbb{E}_X V_i\Big) = 1 - \mathbb{P}\Big(\bigcap_{i\in\mathcal{H}_0} \mathbb{E}_X V_i\Big) = 1 - \prod_{i\in\mathcal{H}_0} \mathbb{P}\big(\mathbb{E}_X V_i\big) = 1 - \prod_{i=6}^{p} \mathbb{P}\big(\mathbb{E}_X V_i\big)$$

The exact value of $\mathbb{P}(V_i)$ and $\mathbb{P}(\mathbb{E}_X V_i)$ depends on the order of p values.

Power

Test statistics: $Z = \frac{\hat{\beta}}{\sigma_{\beta_i}}$. We reject the null hypothesis when $|Z| > \Phi^{-1}(1 - \frac{\alpha}{2})$.

$$power = \mathbb{P}(V_i | H_{0,1}^c) = \mathbb{P}\Big(|Z| > \Phi^{-1}(1 - \frac{\alpha}{2})\Big) = \mathbb{P}\Big(Z > \Phi^{-1}(1 - \frac{\alpha}{2})\Big) + \mathbb{P}\Big(Z < -\Phi^{-1}(1 - \frac{\alpha}{2})\Big)$$

Let's standarize the Z statistics:

$$power = \mathbb{P}\Big(\frac{Z-3}{\sigma_{\beta_i}} > \frac{\Phi^{-1}(1-\frac{\alpha}{2})\sigma_{\beta_i} - 3}{\sigma_{\beta_i}}\Big) + \mathbb{P}\Big(\frac{Z-3}{\sigma_{\beta_i}} < \frac{-\Phi^{-1}(1-\frac{\alpha}{2})\sigma_{\beta_i} - 3}{\sigma_{\beta_i}}\Big)$$

$$power = 1 - \Phi\Big(\frac{\Phi^{-1}(1-\frac{\alpha}{2})\sigma_{\beta_i} - 3}{\sigma_{\beta_i}}\Big) + \Phi\Big(\frac{-\Phi^{-1}(1-\frac{\alpha}{2})\sigma_{\beta_i} - 3}{\sigma_{\beta_i}}\Big)$$

n = 10

$$power = 1 - \Phi\Big(\frac{\Phi^{-1}(1-\frac{\alpha}{2}) - 3}{\sigma_{\beta_i}}\Big) + \Phi\Big(\frac{-\Phi^{-1}(1-\frac{\alpha}{2}) - 3}{\sigma_{\beta_i}}\Big) =$$
$$1 - \Phi\Big(\frac{\Phi^{-1}(1-\frac{\alpha}{2}) - 3}{\frac{1000}{\sqrt{989}}}\Big) + \Phi\Big(\frac{-\Phi^{-1}(1-\frac{\alpha}{2}) - 3}{\frac{1000}{\sqrt{989}}}\Big) \approx 0.84$$

n = 100

$$power = 1 - \Phi\Big(\frac{\Phi^{-1}(1-\frac{\alpha}{2}) - 3}{\sigma_{\beta_i}}\Big) + \Phi\Big(\frac{-\Phi^{-1}(1-\frac{\alpha}{2}) - 3}{\sigma_{\beta_i}}\Big) =$$
$$1 - \Phi\Big(\frac{\Phi^{-1}(1-\frac{\alpha}{2}) - 3}{\frac{1000}{\sqrt{899}}}\Big) + \Phi\Big(\frac{-\Phi^{-1}(1-\frac{\alpha}{2}) - 3}{\frac{1000}{\sqrt{899}}}\Big) \approx 0.81$$

n = 500

$$power = 1 - \Phi\Big(\frac{\Phi^{-1}(1-\frac{\alpha}{2}) - 3}{\sigma_{\beta_i}}\Big) + \Phi\Big(\frac{-\Phi^{-1}(1-\frac{\alpha}{2}) - 3}{\sigma_{\beta_i}}\Big) =$$
$$1 - \Phi\Big(\frac{\Phi^{-1}(1-\frac{\alpha}{2}) - 3}{\frac{1000}{\sqrt{499}}}\Big) + \Phi\Big(\frac{-\Phi^{-1}(1-\frac{\alpha}{2}) - 3}{\frac{1000}{\sqrt{499}}}\Big) \approx 0.64$$

n = 950

$$power = 1 - \Phi\Big(\frac{\Phi^{-1}(1-\frac{\alpha}{2}) - 3}{\sigma_{\beta_i}}\Big) + \Phi\Big(\frac{-\Phi^{-1}(1-\frac{\alpha}{2}) - 3}{\sigma_{\beta_i}}\Big) =$$
$$1 - \Phi\Big(\frac{\Phi^{-1}(1-\frac{\alpha}{2}) - 3}{\frac{1000}{\sqrt{49}}}\Big) + \Phi\Big(\frac{-\Phi^{-1}(1-\frac{\alpha}{2}) - 3}{\frac{1000}{\sqrt{49}}}\Big) \approx 0.78$$

mFDR

$$FDR = \mathbb{E}\frac{V}{max(R,1)}$$

$$mFDR = \frac{\mathbb{E}V}{max(\mathbb{E}R,1)} = \frac{\mathbb{E}\Big[\sum_{i\in\mathcal{H}_0} V_i\Big]}{max\Big(\mathbb{E}\big[\sum_{i=1}^{p} R\big],1\Big)} = \frac{\sum_{i\in\mathcal{H}_0}\mathbb{E}\Big[V_i\Big]}{max\Big(\sum_{i\in\mathcal{H}_0}\mathbb{E}R + \sum_{i\in\mathcal{H}_0^c}\mathbb{E}R, 1\Big)} = \frac{p_0\cdot\alpha}{max\big(p_0\cdot\alpha + (p-p_0)\cdot power, 1\big)}$$

Single simualtion

In this report we will consider data from the model:

$$Y = X\beta + \epsilon$$

where $\beta_1 = \ldots = \beta_5 = 3$, $\beta_6 = \ldots = \beta_{950} = 0$, $X_{i,j} \sim \mathcal{N}(0, \frac{1}{1000})$ and $\epsilon \sim \mathcal{N}(0, I)$.

We will estimate the coefficients using 1000 observations.

```r
alpha <- 0.1
beta_vec <- c(rep(3, 5), rep(0, 945))
X <- matrix(rnorm(1000 * 950, 0, sqrt(1/1000)), nrow=1000)
Y <- X %*% beta_vec + rnorm(1000, 0, 1)

bonferroni <- function(pvals, alpha=0.05) {
  n <- length(pvals)
  pvals < (alpha / n)
}

benjamini_hochberg <- function(pvals, alpha=0.05) {
  n <- length(pvals)
  ord <- order(pvals)
  ord2 <- order(ord)
  res <-( pvals[ord] < ((alpha * seq(n) / n)))
  sapply(1:n, function(i) any(res[i:n]))[ord2]
}

results <- data.frame()

for (i in c(10, 100, 500, 950)) {
  model <- lm(Y~X[, 1:i] - 1)
  msum <- summary(model)
  pvals <- msum$coefficients[, 4]

  # a)
  significance <- pvals < alpha

  # b)
  # TODO: z definicji
  sd_val <- mean(msum$coefficients[, 2])
  CI_len <- mean(abs(confint(model, level=0.9)[, 1] - confint(model, level=0.9)[, 2]))

  # c)
  res_i <- pvals <= alpha
  i_TP <- sum(res_i[1:i] & (beta_vec[1:i] > 0))
  i_FP <- sum(res_i[1:i] & (beta_vec[1:i] == 0))

  res_ii <- bonferroni(pvals, alpha)
  ii_TP <- sum(res_ii[1:i] & (beta_vec[1:i] > 0))
  ii_FP <- sum(res_ii[1:i] & (beta_vec[1:i] == 0))

  res_iii <- benjamini_hochberg(pvals, alpha)
  iii_TP <- sum(res_iii[1:i] & (beta_vec[1:i] > 0))
  iii_FP <- sum(res_iii[1:i] & (beta_vec[1:i] == 0))

  results <- rbind(results, c(sd_val, CI_len, i_TP, i_FP, ii_TP, ii_FP, iii_TP, iii_FP))
}
colnames(results) <-  c("sd", "CI len", "TP", "FP", "Bonf TP", "Bonf FP", "BH TP", "BH FP")
rownames(results) <- c(10, 100, 500, 950)
kable(results, digits = 3)
```

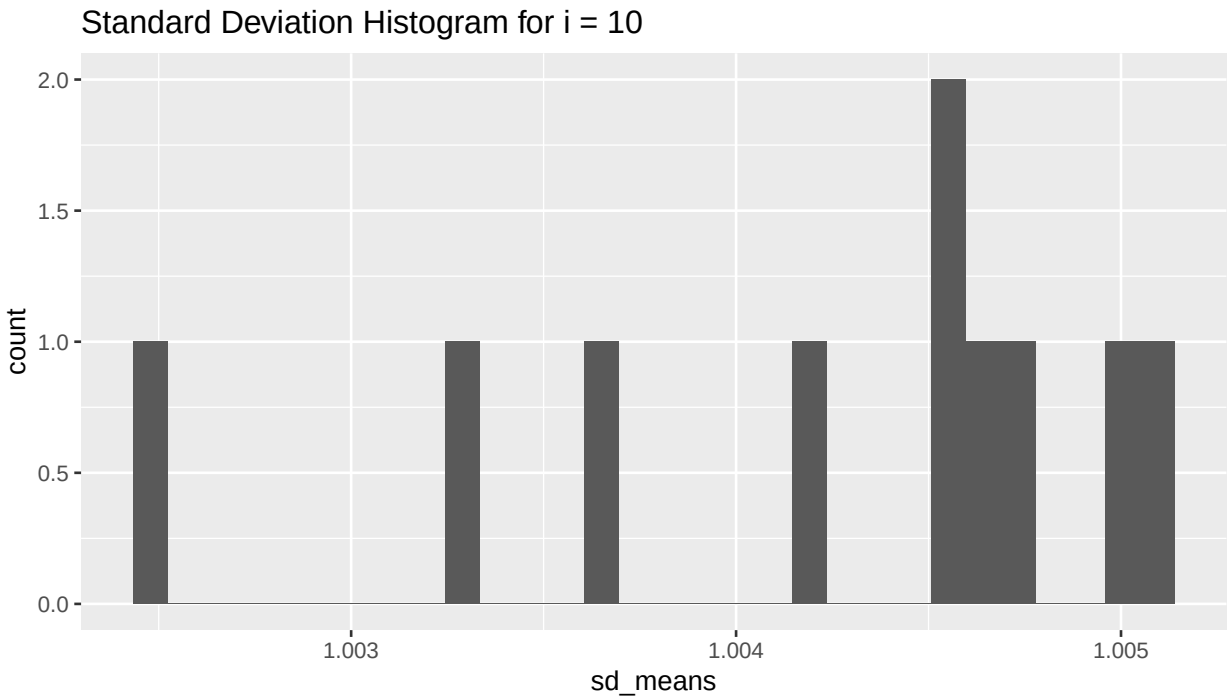
	sd	CI len	TP	FP	Bonf TP	Bonf FP	BH TP	BH FP
10	0.996	3.280	4	0	2	0	3	0
100	1.048	3.452	5	6	1	0	1	0
500	1.398	4.608	2	47	0	0	0	0
950	4.391	14.717	1	105	0	0	0	0

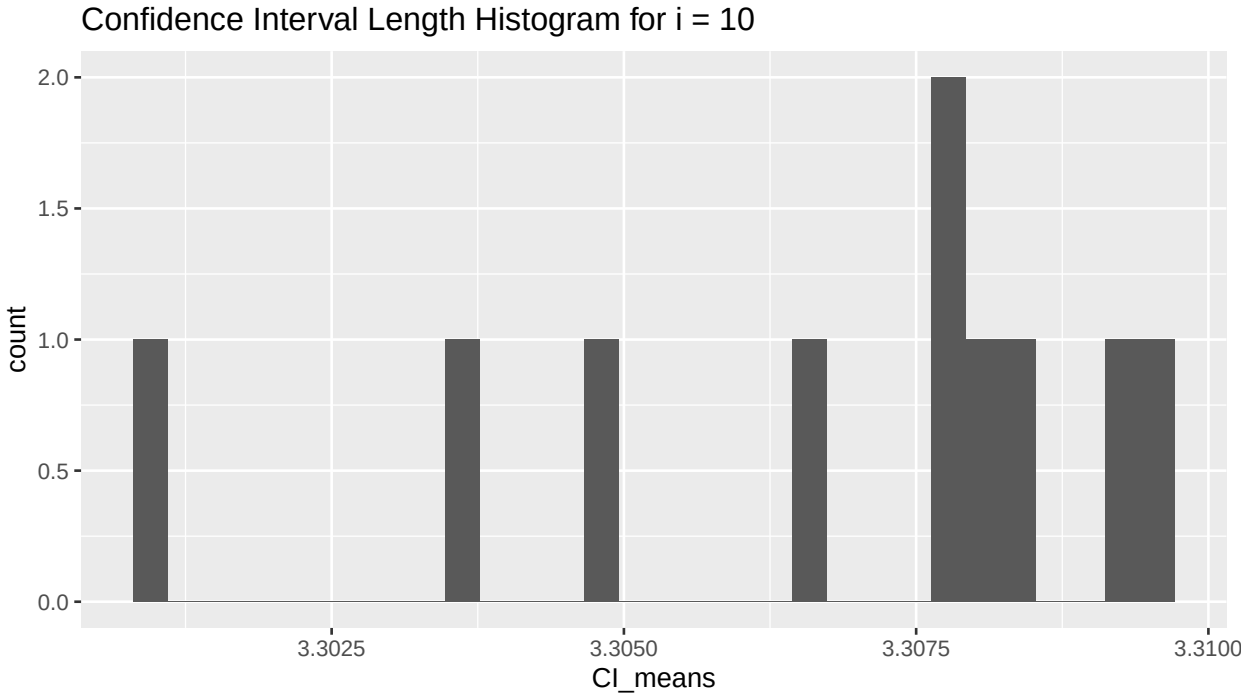
Estimation

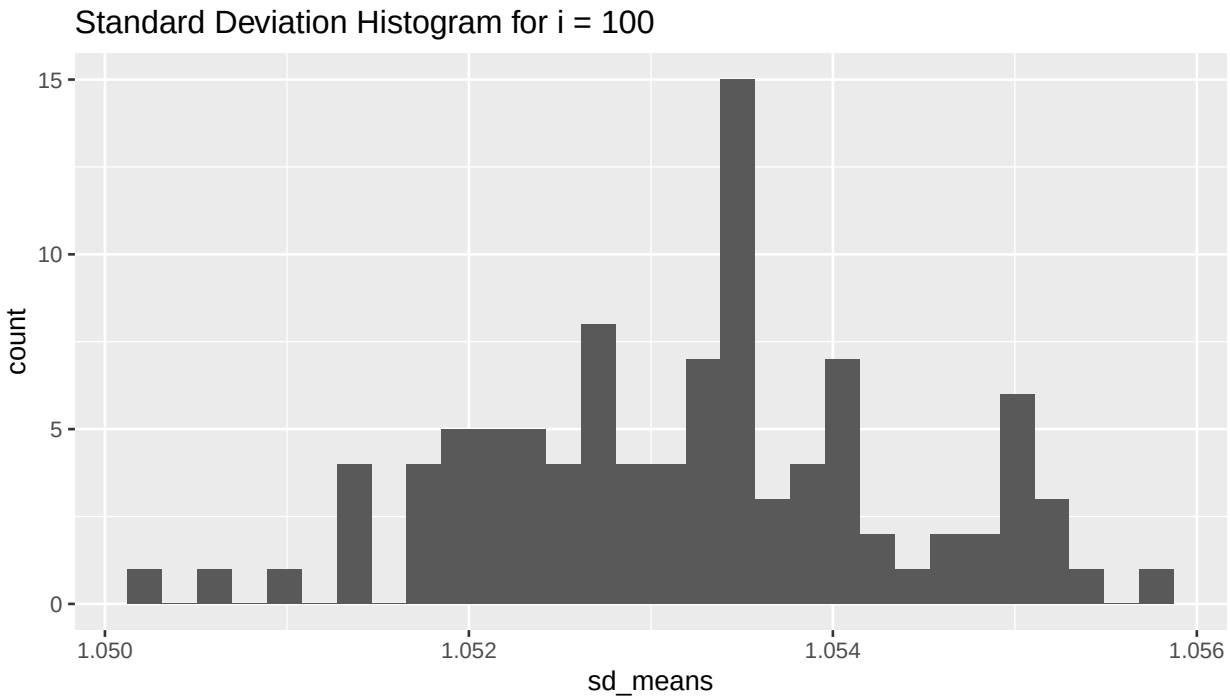
We will run 500 simulations.

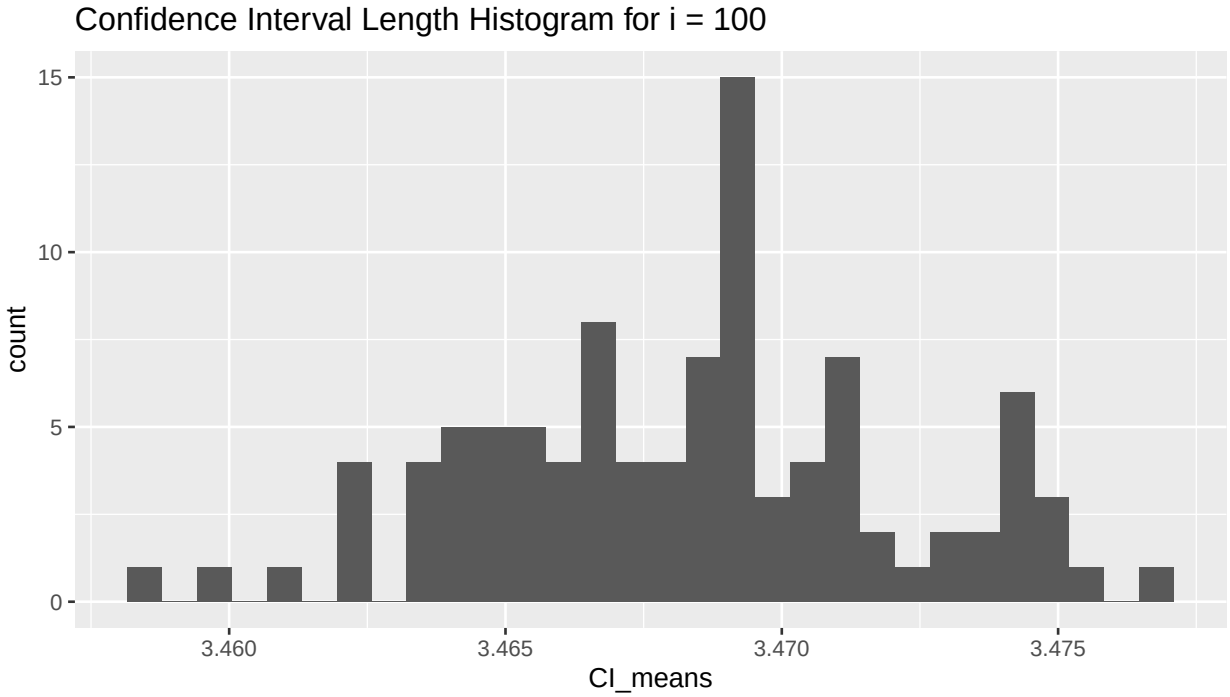
```
FWER <- function(true_values, test_results) {
  # T - reject H0, F - accept H0
  as.integer(any(test_results[which(!true_values)]))
}

FDR <- function(true_values, test_results) {
  sum(test_results[which(!true_values)]) / max(sum(test_results), 1)
}

power <- function(true_values, test_results) {
  mean(test_results[which(true_values)])
}

sim_step <- function(i, X, Y) {
  model <- lm(Y~X[, 1:i] - 1)
  msum <- summary(model)
  # a
  sd_val <- msum$coefficients[, 2]
  # separatly
  # b
  CI_len <- abs(confint(model, level=0.9)[, 1] - confint(model, level=0.9)[, 2])
  # c
  pvals <- msum$coefficients[, 4]
  res_i <- pvals <= alpha
  res_ii <- bonferroni(pvals, alpha)
  res_iii <- benjamini_hochberg(pvals, alpha)
  true_vals <- beta_vec[1:i] > 0

  FWER_i <- FWER(true_vals, res_i)
  FDR_i <- FDR(true_vals, res_i)
  power_i <- power(true_vals, res_i)

  FWER_ii <- FWER(true_vals, res_ii)
  FDR_ii <- FDR(true_vals, res_ii)
  power_ii <- power(true_vals, res_ii)

  FWER_iii <- FWER(true_vals, res_iii)
  FDR_iii <- FDR(true_vals, res_iii)
  power_iii <- power(true_vals, res_iii)

  return(list(sd = sd_val, CI = CI_len,
              FWER_st = FWER_i, FDR_st = FDR_i, power_st = power_i,
              FWER_bonf = FWER_ii, FDR_bonf = FDR_ii, power_bonf = power_ii,
```

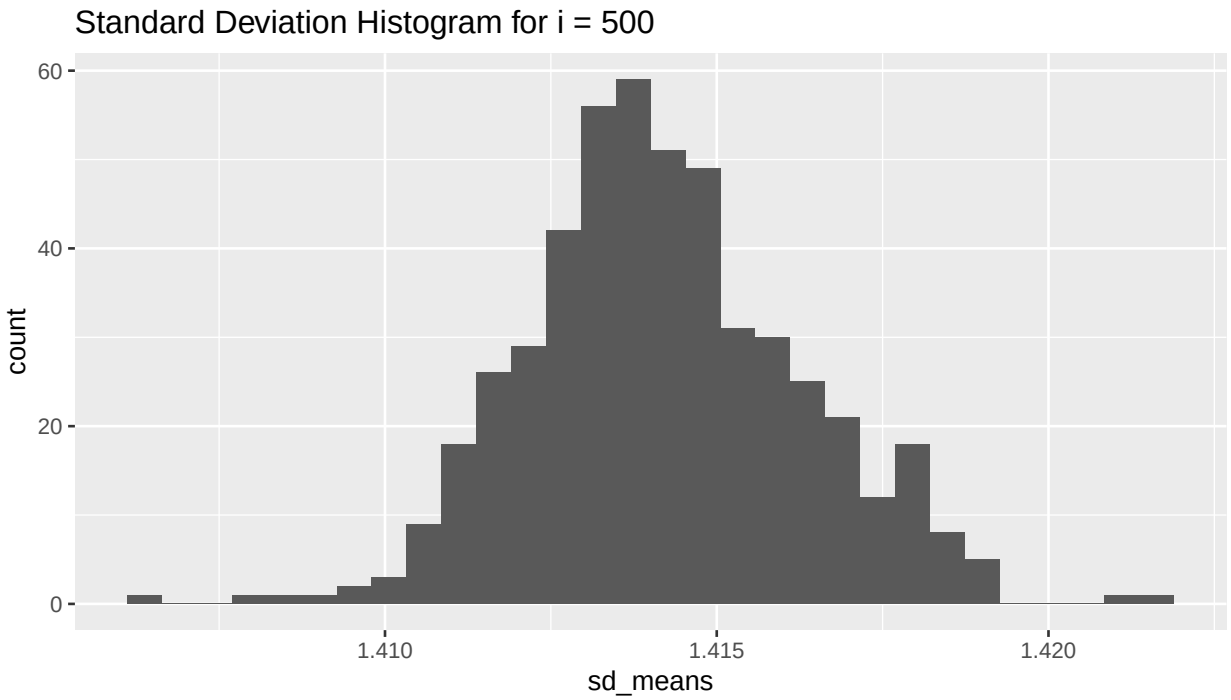
```
              FWER_bh = FWER_iii, FDR_bh = FDR_iii, power_bh = power_iii))
}

i_vec <- c(10, 100, 500, 950)

simulation <- function() {
  X <- matrix(rnorm(1000 * 950, 0, sqrt(1/1000)), nrow=1000)
  Y <- X %*% beta_vec + rnorm(1000, 0, 1)
  sapply(i_vec, function(i) sim_step(i, X, Y))
}

results <- replicate(500, simulation())

tab <- data.frame()

for (i in 1:length(i_vec)) {
  sd_means <- rowMeans(simplify2array(results[1, i, ]))

  print(ggplot() +
    geom_histogram(aes(x=sd_means)) +
      ggtitle(str_c("Standard Deviation Histogram for i = ",
                    as.character(i_vec[i]))))

  CI_means <- rowMeans(simplify2array(results[2, i, ]))

  print(ggplot() +
    geom_histogram(aes(x=CI_means)) +
      ggtitle(str_c("Confidence Interval Length Histogram for i = ",
                    as.character(i_vec[i]))))

  FWERv <- mean(simplify2array(results[3,  i, ]))
  FDRv <- mean(simplify2array(results[4,  i, ]))
  powerv <- mean(simplify2array(results[5,  i, ]))
  FWERb <- mean(simplify2array(results[6,  i, ]))
  FDRb <- mean(simplify2array(results[7,  i, ]))
  powerb <- mean(simplify2array(results[8,  i, ]))
  FWERbh <- mean(simplify2array(results[9,  i, ]))
  FDRbh <- mean(simplify2array(results[10,  i, ]))
  powerbh <- mean(simplify2array(results[11,  i, ]))

  tab <- rbind(tab, c(i_vec[i], mean(sd_means), mean(CI_means), FWERv, FDRv, powerv,
                      FWERb, FDRb, powerb,  FWERbh, FDRbh, powerbh))

}
```

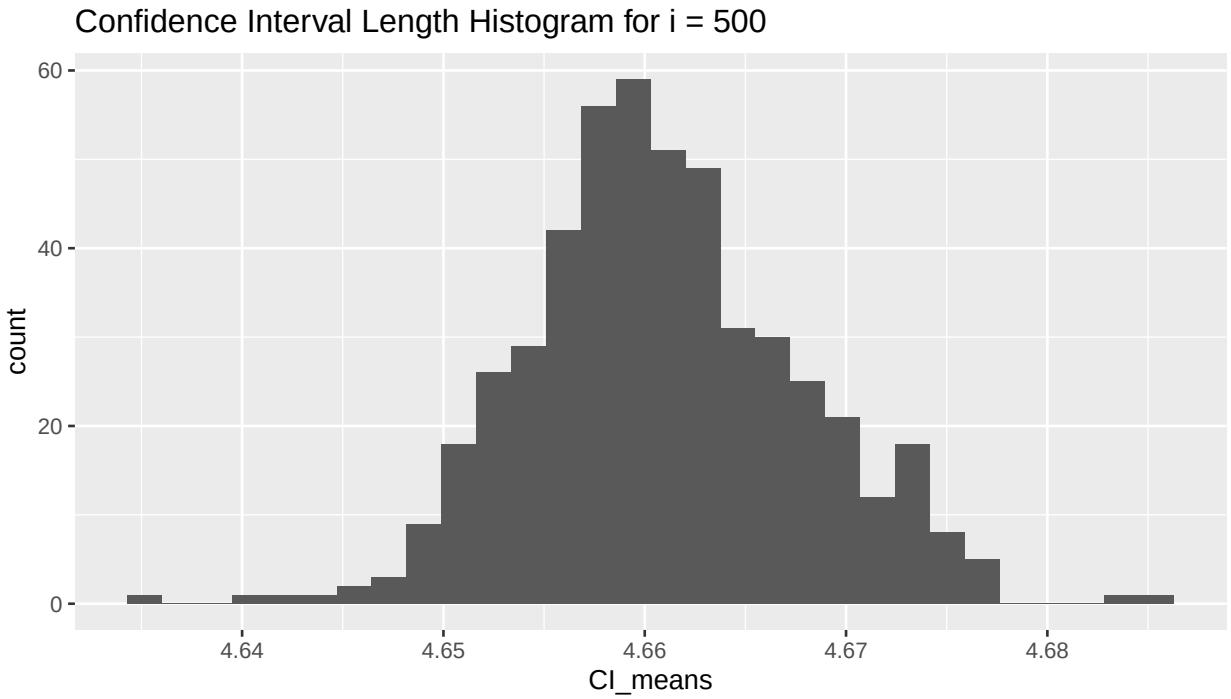

Standard Deviation Histogram for i = 10
count
2.0
1.5
1.0
0.5
0.0
1.003
1.004
1.005
sd_means

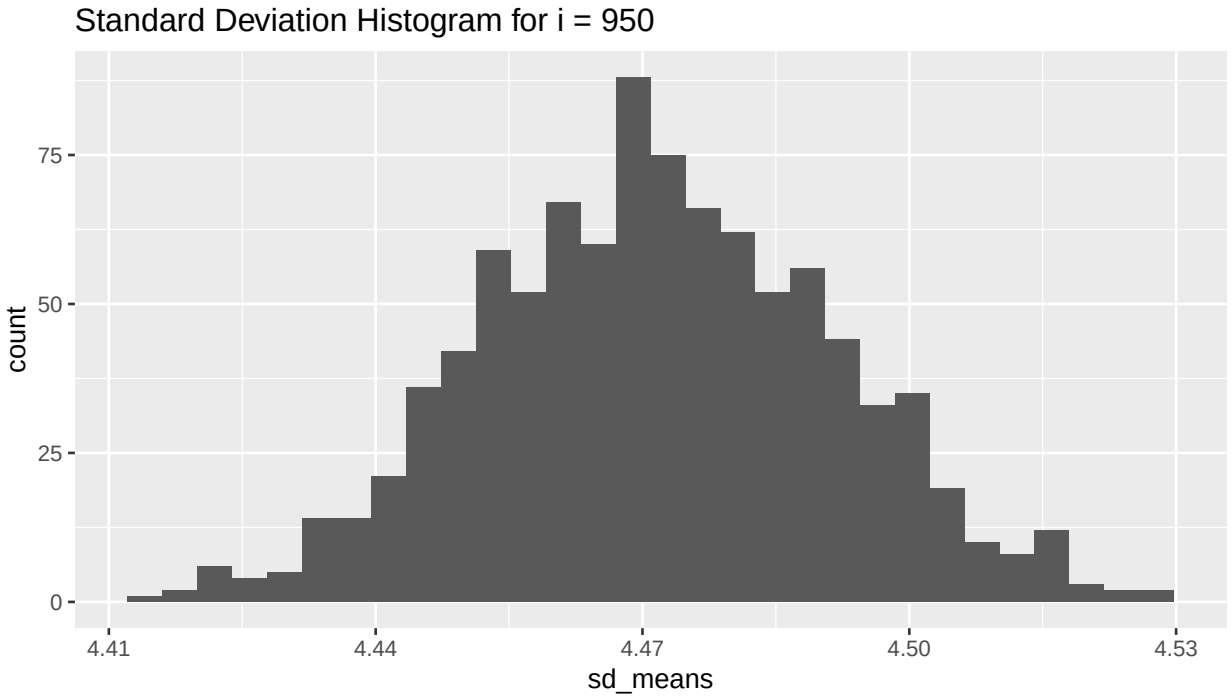

Confidence Interval Length Histogram for i = 10
count
2.0
1.5
1.0
0.5
0.0
3.3025
3.3050
3.3075
3.3100
CI_means

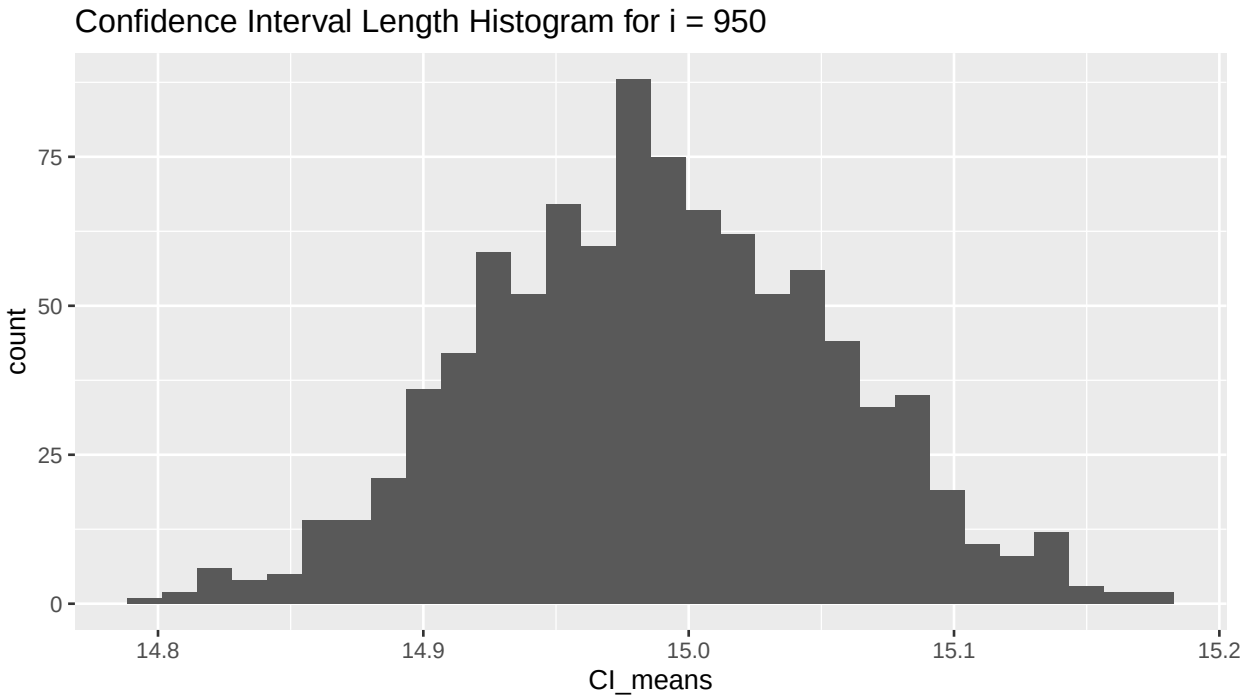

Standard Deviation Histogram for i = 100
15
10
5
0
count
1.050
1.052
1.054
1.056
sd_means


Confidence Interval Length Histogram for i = 100
15
10
5
0
count
3.460
3.465
3.470
3.475
CI_means


Standard Deviation Histogram for i = 500
60
40
20
0
count
1.410
1.415
1.420
sd_means


Confidence Interval Length Histogram for i = 500
60
40
20
0
count
4.64
4.65
4.66
4.67
4.68
CI_means






```
res <-  t(tab)
rownames(res) <- c("i", "sd", "CI", "FWER",  "FDR", "power", "FWER_Bonf",
                   "FDR_Bonf", "power_Bonf",  "FWER_BH", "FDR_BH", "power_BH")
kable(res[c("sd", "CI", "FWER",  "FDR", "power", "FWER_Bonf", "FDR_Bonf",
            "power_Bonf",  "FWER_BH", "FDR_BH", "power_BH"),],
      row.names = T, col.names = c("10", "100", "500", "950"), digits=2)
```

	10	100	500	950
sd	1.00	1.05	1.41	4.47
CI	3.31	3.47	4.66	14.99
FWER	0.40	1.00	1.00	1.00
FDR	0.08	0.67	0.93	0.99
power	0.93	0.90	0.70	0.18
FWER_Bonf	0.06	0.05	0.09	0.05
FDR_Bonf	0.01	0.02	0.07	0.05
power_Bonf	0.67	0.34	0.06	0.00
FWER_BH	0.24	0.24	0.13	0.07
FDR_BH	0.05	0.08	0.09	0.06
power_BH	0.85	0.47	0.07	0.00

As we can see in the above table, adding many irrelevant variables seriously affect the quality of the model. The standard deviation and the confidence intervals are growing. For 945 irrelevant variables the confidence intervals are extremally wide. Thus, in the standard approach we are making very many discoveries. We reject almost all hypotheses. On the other hand, using any correction causes lower power.

To summarise: including many irrelevant variables in the model leads to low-quality model. Presented corrections can provide only limited assistance.