Modele liniowe - raport 4

Łukasz Rębisz

18.01.2023

Zadanie 1

a)

Generujemy macierz $X_{100\times 2}$, której wiersze są niezależnymi wektorami z rozkładu dwuwymiarowego normalnego $N(0, \Sigma/100)$, gdzie

$$\Sigma = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}.$$

Funkcja pozwalająca na przejście ze zmiennej $X \sim N_2(0, I)$ do zmiennej $Y \sim N_2(\mu, \Sigma)$ (pochodząca z Raportu 1):

```
vectors_I_to_sigma <- function(n, mu, sigma){
  # n - liczba wektorów losowych
  # mu - wartość oczekiwana rozkładu Y
  # sigma - macierz kowariancji rozkładu Y

  # Y = A*X+B

  B <- matrix(mu, ncol = 2, nrow = n, byrow=TRUE)
  A <- matrix(c(sqrt(sigma[1,1]), 0, sigma[2,1]/sqrt(sigma[1,1]),
                sqrt(sigma[2,2]-(sigma[2,1]^2)/sigma[1,1])), nrow=2, byrow = TRUE)

  X <- matrix(rnorm(2*n), nrow=n)

  Y <- matrix(nrow=n, ncol = 2)
  Y <- X %*%t(A) + B

  return(Y)
}
```

```
n <- 100
sigma <- matrix(c(1, 0.9, 0.9, 1), byrow = TRUE, nrow = 2)
mu <- c(0,0)
X <- vectors_I_to_sigma(n, mu, sigma/n)
```

Następnie wygenerujmy wektor zmiennej wyjaśnianej $Y = \beta_1 X_1 + \epsilon$, gdzie $\beta_1 = 3$, X_1 to pierwsza kolumna wygenerowanej powyżej macierzy X, natomiast $\epsilon \sim N(0, I)$.

```
beta_1 <- 3
vec_epsilon <- rnorm(n)
Y <- beta_1*X[,1] + vec_epsilon
```

b)

Naszym zadaniem jest skonstruowanie 95% przedziałów ufności dla β_1 dla:

- modelu regresji liniowej prostej $Y = \beta_0 + \beta_1 X_1 + \epsilon$,
- modelu regresji liniowej obejmującego obie zmienne wyjaśniające $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$.

W przypadku **prostej regresji liniowej**:

```
data <- data.frame(Y,X[,1],X[,2])
colnames(data) <- c("Y", "X1", "X2")
simple_lin_mod <- lm(Y~X1, data)
simple_conf_int <- c(confint(simple_lin_mod)[2,][["2.5 %"]],
                     confint(simple_lin_mod)[2,][["97.5 %"]])
```

95% przedział ufności dla β_1: [0.988, 4.817], długość przedziału ufności 3.8290918.

W przypadku **wielorakiej regresji liniowej**:

```
multiple_lin_mod <- lm(Y~X1+X2, data)
multiple_conf_int <- c(confint(multiple_lin_mod)[2,][["2.5 %"]],
                       confint(multiple_lin_mod)[2,][["97.5 %"]])
```

95% przedział ufności dla β_1: [0.097, 9.682], długość przedziału ufności 9.585014.

Oba otrzymane przedziały ufności zawierają teoretyczna wartość $\beta_1 = 3$, jednak długość przedziału ufności otrzymanego metodą **prostej** regresji liniowej jest blisko **dwa razy mniejsza**. Otrzymany wynik jest zgodny z przewidywaniami, ponieważ w przypadku regresji liniowej wielorakiej popełniany błąd (więc także wariancja) ma źródło zarówno w przyjętym błędzie losowym ϵ, jak i we wszystkich zmiennych objaśniających. Jasno obrazują to poniższe wzory porównujące konstrukcję przedziału ufności dla paramteru β_1 w obu przyjętych modelach:

Wzory teoretyczne

$$H_0 : \beta_1 = 0 \quad \text{vs} \quad H_1 : \beta_1 \neq 0$$

W przypadku **regresji liniowej wielorakiej**, chcąc przetestować istotność pojedynczego parametru β_1, statystyka testowa ma postać:

$$T_1 = \frac{\hat{\beta}_1}{s(\hat{\beta}_1)} \sim t(n-p), \quad \text{gdzie} \quad p - \text{liczba parametrów modelu}, \quad s^2(\hat{\beta}_1) = s^2(\mathbb{X}'\mathbb{X})^{-1}_{2,2}$$

Zatem przedział ufności dla β_1 to w tym przypadku:

$$\hat{\beta}_1 \pm t^{-1}(1-\alpha/2, n-3) \cdot s(\hat{\beta}_1)$$

Estymator $\hat{\beta}_1$ pochodzi z macierzy $\hat{\beta} = (\mathbb{X}'\mathbb{X})^{-1}\mathbb{X}'Y$.

$$s^2(\hat{\beta}) = s^2(\mathbb{X}'\mathbb{X})^{-1} = \frac{1}{n-3}||e||_2^2 = \frac{1}{n-3}\sum_{i=1}^{n}(Y_i - (\mathbb{X}_{i1}\hat{\beta}_0 + \mathbb{X}_{i2}\hat{\beta}_1 + \mathbb{X}_{i3}\hat{\beta}_2))^2$$

Natomiast w przypadku **regresji liniowej prostej** przedział ufności dla β_1 ma postać:

$$\hat{\beta}_1 \pm t^{-1}(1-\alpha/2, n-2) \cdot s(\hat{\beta}_1)$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^{n}(X_i - \bar{X})^2}, \qquad s^2(\hat{\beta}_1) = \frac{s^2}{\sum_{i=1}^{n}(X_i - \bar{X})^2},$$

ponieważ statystyka testowa

$$T = \frac{\hat{\beta}_1 - \beta_1}{s(\hat{\beta}_1)} \sim t(n-2).$$

Testy istotności:

Porównajmy wyniki t-testów (zgodnych z powyższymi wzorami) na poziomie istotności $\alpha = 0.05$ dla obu modeli:

	Statystyka T	p-wartość
Model prosty	9.048621	0.0033410
Model wieloraki	4.918539	0.0456288

Otrzymana wartość statystyki testowej T jest około **dwa razy większa** w przypadku **prostej** regresji liniowej. Oznacza to, że w modelu prostym wpływ zmiennej X_1 na zmienną objaśnianą jest większy niż w przypadku modelu wielorakiego.

Porównanie p-wartości dla obu testów pokazuje natomiast, że w przypadku modelu prostego odrzucilibyśmy hipotezę zerową na rzecz alternatywy (tzn. $\beta_1 \neq 0$) z prawdopodobieństwem 0.997.

Natomiast w modelu wielorakim nie mamy podstaw do odrzucenia hipotezy zerowej (tzn. $\beta_1 = 0$), ponieważ p-wartość wynosi aż 0.046.

c)

Wyznaczmy **odchylenie standardowe** estymatora β_1 w obu modelach.

Regresja liniowa prosta:

```
beta_1 <- function(X, Y){
  return(sum((X-mean(X))*(Y-mean(Y)))/sum((X-mean(X))^2))
}

beta_0 <- function(X, Y){
  return(mean(Y) - beta_1(X, Y)*mean(X))
}

s2 <- function(X, Y){
  n <- length(X)
  b_0 <- beta_0(X,Y)
  b_1 <- beta_1(X,Y)

  return((1/(n-2))*sum((Y-b_0-b_1*X)^2))
}

s_simple <- sqrt(s2(data$X1, data$Y)/((n-1)*var(data$X1)))
```

Odchylenie standarodowe $s(\hat{\beta}_1) = 0.9648$.

Regresja wieloraka:

```
multiple_s2 <- function(X, Y){
  n <- dim(X)[1]
  p <- dim(X)[2]

  betas <- solve(t(X)%*%X)%*%t(X)%*%Y
```

```
  return((1/(n-p))*sum((Y-(X%*%betas))^2))
}

mul_s2 <- multiple_s2(X,Y)
s_multiple <- sqrt(mul_s2*solve(t(X)%*%X)[2,2])
```

Odchylenie standarodowe $s(\hat{\beta}_1) = 2.3712$.

Zatem odchylenie standardowe $s(\hat{\beta}_1)$ jest około **dwa razy większe** w przypadku modelu **wielorakiej** regresji liniowej. Obserwacja jest analogiczna do otrzymanych powyżej wyników dotyczących przedziałów ufności dla paramteru β_1. Także obrazuje wpływ dodatkowej zmiennej objaśniającej X_2 na wielkość generowanych błędów.

Wyznaczmy **teoretyczną moc** identyfikacji estymatora β_1 w obu modelach (tzn. prawdopodobieństwo odrzucenia hipotezy $H_0 : \beta_1 = 0$):

$$\gamma(\beta_1) = P_{H_1}(|T| > t^{-1}(1-\alpha/2, \quad df, \quad ncp = 0)) =$$
$$1 - P_{H_1}(|T| \leq t^{-1}(1-\alpha/2, \quad df, \quad ncp = 0)) =$$
$$1 - P_{H_1}(T \leq t^{-1}(1-\alpha/2, \quad df, \quad ncp = 0)) + P_{H_1}(T \leq t^{-1}(\alpha/2, \quad df, \quad ncp = 0)),$$

gdzie liczba stopni swobody $ncp = n-2$ w przypadku modelu prostego oraz $ncp = n-p = n-3$ w przypadku modelu wielorakiego. Statystka $T \sim t(df, ncp = \beta_1 = 3/s(\widehat{\beta_1}))$.

```
alpha <- 0.05
power_simple <- 1-pt(qt(1-alpha/2, df=n-2,ncp=0), df=n-2,ncp=3/s_simple) +pt(qt(alpha/2,
                                                  df=n-2,ncp=0), df=n-2,
                                                  ncp=3/s_simple)

power_multiple <- 1-pt(qt(1-alpha/2, df=n-3,ncp=0), df=n-3,ncp=3/s_multiple) +pt(qt(alpha/2,
                                                    df=n-3,ncp=0), df=n-3,
                                                    ncp=3/s_multiple)
```

Teretyczna moc testu, przy prawdziwej alternatywie $\beta_1 = 3$, wynosi:

- 0.868 w przypadku regresji prostej,
- 0.24 w przypadku regresji wielorakiej.

d)

Powtórzmy powyższe obliczenia dla 1000 niezależnych wektorów błędu losowego ϵ. Dla każdego wektora ϵ obliczmy wektor zmiennej odpowiedzi $Y = \beta_1 X + \epsilon$.

Następnie wykonajmy dla otrzymanych 1000 kopii danych estymację parametru β_1, test istotności dla tego parametru w przypadku modelu regresji prostej i wielorakiej.

```
M <- 1000
epsilons <- matrix(rnorm(M*n), nrow=M)
beta_1 <- 3

Y <- epsilons + rep(beta_1*X[,1], each = nrow(epsilons))
```

```
beta_1_t_test <- function(k){
  Y_new <- Y[k,]
  new_data <- data.frame(Y_new,X[,1],X[,2])
  colnames(new_data) <- c("Y", "X1", "X2")
```

```
  simple_lm <- lm(Y~X1, new_data)
  multiple_lm <- lm(Y~X1+X2, new_data)

  beta_1 <- c(summary(simple_lm)$coefficients[2,1],
              summary(multiple_lm)$coefficients[2,1])

  p_value <-  c(summary(simple_lm)$coefficients[2,4],
                summary(multiple_lm)$coefficients[2,4])

  reject <- c(p_value[1] < alpha, p_value[2] < alpha)

  return(c(beta_1[1], beta_1[2], p_value[1], p_value[2], reject[1], reject[2]))
}

vec_beta_1_t_test <- Vectorize(beta_1_t_test)
```

	beta_1	p-wartość	moc testu	moc teor.	s prób.	s teor.
Model prosty	3.020512	0.0227628	0.883	0.868	0.9288729	0.9647656
Model wieloraki	3.003571	0.2999961	0.228	0.240	2.4321188	2.3712359

Analiza powyższych średnich wyników (dla 1000 powtórzeń eksperymentu) pozwala stwierdzić, że:

- w obu modelach wyestymowana wartość parametru $\hat{\beta}_1$ jest bliska wartości teoretycznej równej $\beta_1 = 3$,
- p-wartość dla modelu prostego pozwala odrzucić hipotezę zerową mówiącą o nieistotności paramteru β_1 z prawdopodobieństwem 0.9772372, natomiast w przypadku modelu wielorakiego nie udrzucilibyśmy hipotezy zerowej z powodu dużej p-wartości,
- wyestymowana moc testu dla modelu prostego jest bliska mocy teoretycznej równej 0.868, moc dla modelu wielorakiego jest również bliska mocy teoretycznej równej 0.24,
- model wieloraki charakteryzuje się ok. dwukrotnie większą wariancją. Otzymane wyniki dla wariancji są bliskie wartościom teoretycznym.

Wyniki otrzymane dla obliczeń teoretycznych i empirycznych są zgodne (wyniki dla mocy testu istotności parametru β_1 oraz odchylenia $s(\hat{\beta}_1)$.

Powyższe obserwacje wynikają z tego, że zmienne X_1 i X_2 sa mocno skorelowane. Dla przypomnienia macierz Σ to:

$$\Sigma = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}.$$

Zatem korelacja pomiędzy zmiennymi X_1 i X_2 wynosi aż 0.9. Z tego powodu nie jest spełnione założenie modelu regresji liniowej mówiące o niezależności zmiennych objaśniających. Silna zależność pomiędzy X_1 i X_2 wpływa istotnie zwłaszcza na moc testu. Szum pochodzący od zmiennej X_2 istotnie zaburza informacje o zmiennej X_1.

Zadanie 2

a)

Wygenerujmy macierz $X_{1000\times 950}$, której elementy są niezależnymi zmiennymi z rozkładu $N(0, \sigma = 0.1)$. Następnie stwórzmy wektor zmiennych odpowiedzi $Y = X\beta + \epsilon$, gdzie $\beta = (3, 3, 3, 3, 3, 0, ..., 0)^T$.

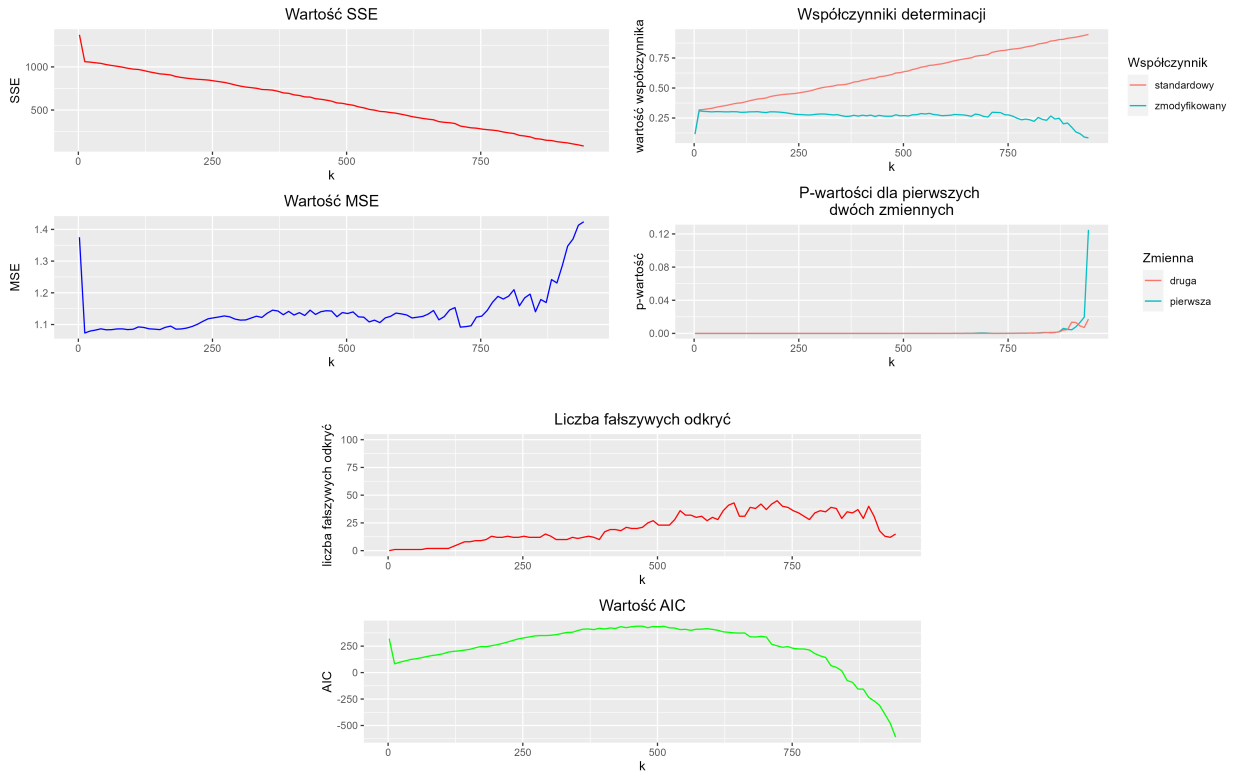
```
alpha = 0.05
n <- 1000
m <- 950

X <- matrix(rnorm(n*m, sd=0.1), nrow=n)
betas <- rep(0, m)
betas[1:5] <- 3

epsilons <- rnorm(n)

Y <- X%*%betas + epsilons
```

b)

Wyznaczmy dla k **pierwszych kolumn** stworzonej macierzy ($1 \leq k \leq 950$) następujące wartości:

- $SSE = ||Y - \hat{Y}||^2$,
- $MSE = \frac{SSE}{dfE} = \frac{SSE}{n-k}$,
- wartość Kryterium informacyjnego Akaikego $AIC = n \log(SSE/n) + 2k$,
- p-wartości dla testów istotności pierwszych dwóch zmiennych objaśniających,
- liczbę błędnych odkryć zmiennych objaśniających (tylko pierwszych 5 zmiennych jest niezerowa).

```
regression_2b <- function(k){
  n <- length(Y)
  X_new <- X[,1:k]
  betas_new <- betas[1:k]
  LinMod <- lm(Y~X_new-1)
  Y_hat <- as.numeric(predict(LinMod))

  SSE <- sum((Y-Y_hat)^2)
  MSE <- SSE/(n-k)
  AIC <- n*log(SSE/n) + 2*k
  p_value_1 <- summary(LinMod)$coefficients[1,4]
  p_value_2 <- summary(LinMod)$coefficients[2,4]
  R_sq <- summary(LinMod)$r.squared
  adj_R_sq <- summary(LinMod)$adj.r.squared

  false_discover <- 0
  if(k>5) false_discover <- sum(as.numeric(summary(LinMod)$coefficients[6:k,4])
                                <alpha)

  return(c(SSE, MSE, AIC, p_value_1, p_value_2, R_sq, adj_R_sq, false_discover))
}

vec_regression_2b <- Vectorize(regression_2b)

vec_k <- seq(2,950, 10)
results_2b <- vec_regression_2b(vec_k)
str(results_2b)
```



Analiza powyższych wykresów pozwala stwierdzić, że:

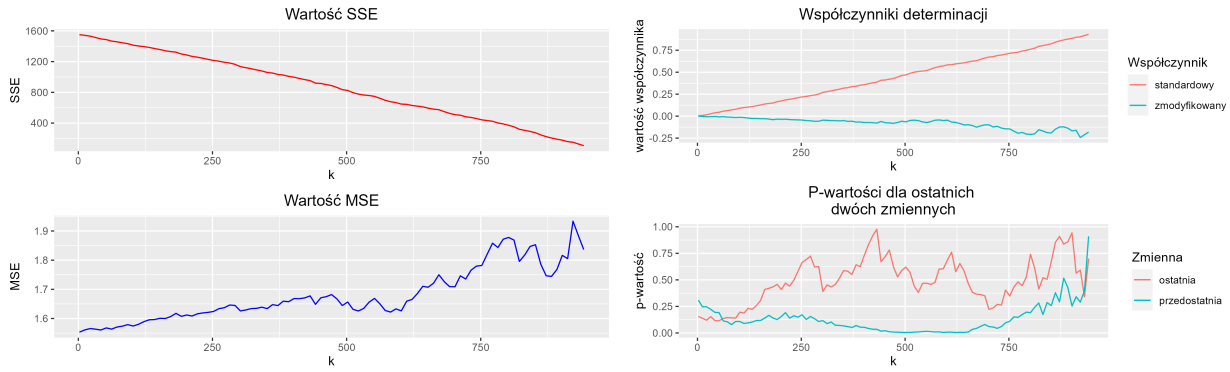
- wartość *SSE* maleje wraz ze wzrostem liczby kolumn. Nie jest to zasakakujące, ponieważ $SSE = \sum_{i=1}^{k}(Y_i - \hat{Y}_i)^2$ - liczba kolumn ma kluczowe znaczenie dla wielkości sumy, mimo różnych estymacji $\hat{Y}$ w zależności od k.
- Prawdziwy charakter błędów lepiej oddaje wartość *MSE*, która zawiera korektę na rozmiar badanej próby (poprzez dzielenie przez $n - k$). W tym przypadku najmniejsze wartości są osiągane dla $k < 250$. Dla skrajnie małej i dużej liczby kolumn k wartości *MSE* są znaczne.
- W podobny sposób zachowują się współczynniki determinacji - współczynnik R^2 jest wrażliwy na rozmiar próby - w tym przypadku stale rośnie wraz ze wzrostem k. Dokładniejszą informację niesie zmodyfikowany współczynnik R^2 uwzględniający rozmiar próby. Przyjmuje wartości bliskie 0.25 bez względu na liczbę kolumn k (poza skrajnymi wartościami k). Wartość ta świadczy o niskim dopasowaniu modelu liniowego do danych.
- Analiza p-wartości dla (istotności) pierwszych dwóch zmiennych wyjaśniających pokazuje, że wartości są stale niskie (poza skrajnie dużymi k) - świadczą o bardzo wysokim prawdopodobieństwie odrzucenia (błędnej) hipotezy mówiącej o braku wpływu tych zmiennych na zmienną objaśnianą.
- Liczba fałszywych odkryć rośnie wraz ze wzrostem próby dla $k < 500$. Następnie stabilizuje się na poziomie ok. 30-40 fałszywych odkryć.
- Najważniejszą informację świadczącą o dokładności modelu niesie wartość kryterium *MSE*. Im mniejsza wartość tego kryterium, tym model jest dokładniejszy. Zatem najdokładniejsze modele otrzymaliśmy dla $k < 250$ oraz $k > 750$.

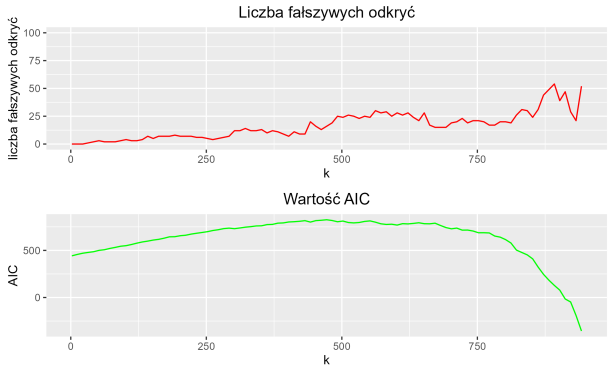
Analiza powyższych wyników pokazuje, jak duży wpływ na dokładność modelu ma liczba badanych kolumn. Skoro tylko pierwszych pięć zmiennych objaśnianych jest niezerowa, to dodanie kolejnych (w rzeczywistości zerowych) kolumn zwiększa generowany błąd. Zatem pomimo tego, że teoretycznie pełen model niesie najlepszą informację o danych, to w takim przypadku doszło jednak to tzw. przeuczenia.

c)

Wykonajmy analogiczną analizę do powyższej, biorąc tym razem k **ostatnich kolumn macierzy**.

```
regression_2c <- function(k){
  n <- length(Y)
  m <- dim(X)[2]
  X_new <- X[,(m-k+1):m]
  betas_new <- betas[(m-k+1):m]
  LinMod <- lm(Y~X_new-1)
  Y_hat <- as.numeric(predict(LinMod))

  SSE <- sum((Y-Y_hat)^2)
  MSE <- SSE/(n-k)
  AIC <- n*log(SSE/n) + 2*k
  p_value_1 <- summary(LinMod)$coefficients[k,4]
  p_value_2 <- summary(LinMod)$coefficients[k-1,4]
  R_sq <- summary(LinMod)$r.squared
  adj_R_sq <- summary(LinMod)$adj.r.squared


  false_discover <- sum(as.numeric(summary(LinMod)$coefficients[1:(min(k,m-5)),4])
                                <alpha)

  return(c(SSE, MSE, AIC, p_value_1, p_value_2, R_sq, adj_R_sq, false_discover))
}

vec_regression_2c <- Vectorize(regression_2c)

vec_k <- seq(2,950, 10)
results_2c <- vec_regression_2c(vec_k)
```





Analiza powyższych wykresów pozwala stwierdzić, że:

- Wzrost *SSE* oraz spadek *MSE* wraz ze wzrostem k są naturalne.
- Ujemny zmodyfikowane współczynnik determinacji jest zaskakujący.
- Obliczanie p-wartości dla ostatnich dwóch zmiennych (zerowych) pokazuje, że ostatnia zmienna zostałaby odrzucena praktycznie bez względu na k, natomiast przedostatnia zmienna zostałaby (błędnie) uznana za najprawdopodobniej istotną dla $k < 700$.
- Liczba fałszywych odkryć rośnie wraz ze wzrostem próby.
- Wartość kryterium *AIC* świadczy o tym że najdokładniejsze modele otrzymaliśmy dla prawie pełnych prób.

Wnioski: w tym przypadku prawdziwie niezerowe zmienne znajdowały się na końcu próby. Z tego powodu ich istotność została stłumiona przez pozostałe, zerowe zmienne. W tym przypadku również doszło do tzw. przeuczenia. Trudno natomiast dostrzec ich istotność, bowiem w momencie pojawienia się rozmiar próby jest już bardzo duży (bardzo duży wpływ błędu zerowych zmiennych).

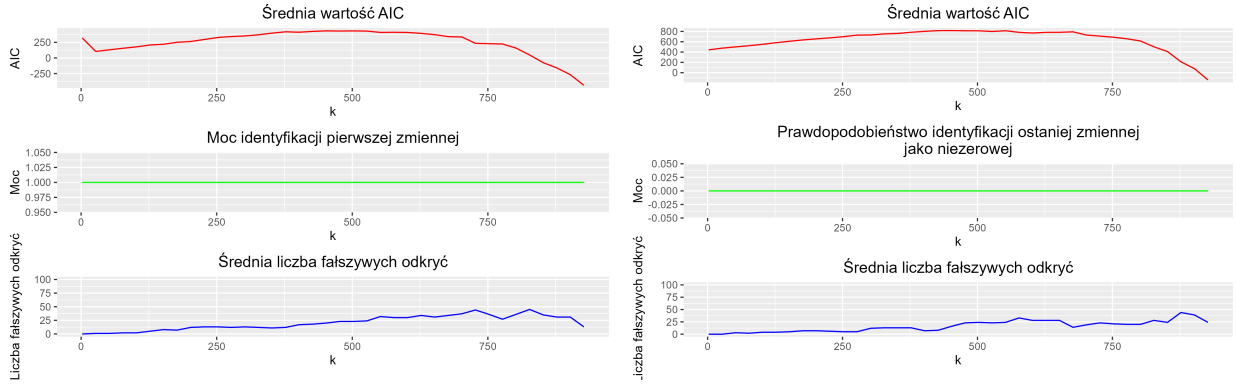
d)

Powtórzmy 1000-krotnie powyższe doświadczenia, za każdym razem wyznaczając:

- moc rozpoznania zmiennej X_1 jako istotnej (niezerowej),
- średnią liczbę fałszywych odkryć dla danego k,
- średni rozmiar próby wybrany przez kryterium *AIC*.

```
regression_2d1 <- function(k){
  n <- length(Y)
  X_new <- X[,1:k]
  betas_new <- betas[1:k]
  LinMod <- lm(Y~X_new-1)
  Y_hat <- as.numeric(predict(LinMod))

  SSE <- sum((Y-Y_hat)^2)
  AIC <- n*log(SSE/n) + 2*k
  p_value_1 <- summary(LinMod)$coefficients[1,4]
  reject_1 <- p_value_1 < alpha

  false_discover <- 0
  if(k>5) false_discover <- sum(as.numeric(summary(LinMod)$coefficients[6:k,4])
                                <alpha)
```

```
  return(c(AIC, reject_1, false_discover))
}

vec_regression_2d1 <- Vectorize(regression_2d1)


regression_2d2 <- function(k){
  n <- length(Y)
  m <- dim(X)[2]
  X_new <- X[,(m-k+1):m]
  betas_new <- betas[(m-k+1):m]
  LinMod <- lm(Y~X_new-1)
  Y_hat <- as.numeric(predict(LinMod))

  SSE <- sum((Y-Y_hat)^2)
  AIC <- n*log(SSE/n) + 2*k
  p_value_1 <- summary(LinMod)$coefficients[k,4]
  reject_1 <- p_value_1 < alpha

  false_discover <- sum(as.numeric(summary(LinMod)$coefficients[1:(min(k,m-5)),4])
                                <alpha)

  return(c(AIC, reject_1, false_discover))
}

vec_regression_2d2 <- Vectorize(regression_2d2)
```



Wykresy po **lewej stronie** przedstawiają średnie wyniki dla 1000 krotnego powtórzenia eksperymentu, gdy brane jest k **pierwszych kolumn** macierzy, natomiast po **prawej stronie** k **ostatnich kolumn**.

Wnioski z powyższych wykresów:

- porównanie wykresów średniej wartości AIC wskazuje, że w obu przypadkach za najdokładniejszy zostałby uznany pełen model (minimalna wartość AIC uzyskiwana dla maksymalnego k). W przypadku gdy brane jest k pierwszych kolumn zauważamy spadek wartości AIC dla początkowych k spowodowany występowaniem niezerowych zmiennych $X_1, ..., X_5$.
- Moc identyfikacji pierwszej zmiennej jako niezerowej jest stale równa 1 - za każdym razem odrzucona została hipoteza $H_0 : \beta_1 = 0$.
- Z kolei prawdopodobieństwo identyfikacji ostatniej (zerowej) kolumny jako niezerowej wynosi stale 0.
- W obu przypadkach średnia liczba fałszywych odkryć stale rośnie. Wyraźny spadek fałszywych odkryć

następuje dopiero dla $k > 900$ (w przypadku modeli, które są prawie pełne).

Powyższe wyniki wyraźnie pokazują, że **najdokładniejsze są modele pełne**. W przypadku dodawania kolejnych zerowych kolumn ogólnie NIE otrzymujemy dokładniejszych modeli. Wzrost wartości AIC oraz średniej liczby fałszywych odkryć wraz ze wzsrostem k (poza skrajnie dużymi k) wyraźnie pokazuje spadek dokładności modelu. Świadczy to ponownie o tzw. przeuczeniu.

Otrzymując dane, o których nie mamy żadnych dodatkowych informacji (w tym przypadku nie wiemy, że tylko pierwszych pięć kolumn jest niezerowa), powinniśmy dokonać analizy pełnego modelu. Otrzymamy wówczas najlepsze dopasowanie do danych. Wybiórczy wybór analizowanych kolumn nie charakteryzuje się zależnością: dokładność modelu wzrasta wraz ze wzrostem liczby kolumn.