Sesja komputerowa 2: R

1. Dla celów analizy potrzebujemy pakietów *foreign*, *survival* i *rms*. Inicjalizujemy je wykonując trzy poniższe polecenia:

```
library(foreign)
library(survival)
library(rms)
```

2. Będziemy używać danych dotyczących czasu przeżycia 102 chorych z NSCLC. By wczytać dane z pliku w formacie STATA, używamy funkcji **read.dta**:

```
nsclc <- read.dta("t:/burzykowskit/biostat/datasets/nsclc_eng.dta")
```

Zawartość obiektu **nsclc** możemy sprawdzić używając funkcji **head**:

```
head(nsclc)
```

3. Do oszacowania modelu PH można użyć funkcji **coxph** z pakietu *survival*. Opis funkcji możemy uzyskać wydając polecenie **help(coxph)**. Na początku rozważymy model z ekspresją białka P53. Odpowiednie polecenie to:

```
nsclc.PH.P53e <- coxph(Surv(survtime, survind) ~ expression, data=nsclc)
```

Obiekt **nsclc.PH.P53e** zawiera szczegóły oszacowanego modelu. **Surv(survtime, survind)** definiuje obiekt klasy *Surv*, z czasem przeżycia i wskaźnikiem zdarzeń zdefiniowanymi zmiennymi **survtime** i **survind**, których wartości pobierane są z obiektu **nsclc**. Syntax **Surv(survtime, survind) ~ expression** wskazuje, że czas przeżycia jest modelowany przy użyciu tylko jednej zmiennej niezależnej, **expression**. Rezultaty modelu możemy uzyskać wydając polecenie:

```
print(nsclc.PH.P53e)
```

Odpowiadają wynikom przedstawionym na slajdzie 34. W celu uzyskania wyniku testu „score", musimy skorzystać z atrybutu **nsclc.PH.P53e$score** obiektu **nsclc.PH.P53e**, który zawiera wartość statystyki testowej. Aby obliczyć poziom krytyczny testu, użyjemy funkcji dystrybuanty rozkładu chi-kwadrat z jednym stopniem swobody:

```
1-pchisq(nsclc.PH.P53e$score,1)
```

Na podstawie modelu możemy oszacować bazową funkcje przeżycia i wynikające stąd oszacowania dla pacjentów z ekspresją białka i bez. Odpowiednie polecenie to:

```
nsclc.pKM.P53e <- survfit(nsclc.PH.P53e, newdata=nsclc)
```

W poleceniu funkcji **survfit** używamy obiektu **nsclc.PH.P53e** z wynikami dla oszacowanego modelu PH. Argument **newdata** wskazuje, że szacujemy funkcje przeżycia dla zbioru danych użytego do szacowania modelu.

Porównamy oszacowania uzyskane na podstawie modelu z oszacowaniami uzyskanymi przy użyciu metody Kaplana-Meiera. Te ostatnie dostajemy wykonując polecenie

```
nsclc.KM.P53e <- survfit(Surv(survtime, survind) ~ expression, data=nsclc)
```

Następnie wykreślamy wyznaczone krzywe przeżycia i dodajemy do wykresu oszacowania uzyskane na podstawie modelu:

```
plot(nsclc.KM.P53e,conf.int=FALSE)
lines(nsclc.pKM.P53e,col=c("red"),mark.time=FALSE)
```

W poleceniu funkcji **lines** używamy obiektu **nsclc.pKM.P53e** z oszacowaniami funkcji przeżycia dla modelu PH, ustalamy kolor odpowiednich krzywych na czerwony i wyłączamy zaznaczanie czasów zdarzeń (**mark.time=FALSE**).

4. Rozważmy model PH z ekspresją białka P53 i mutacją genu P53. W tym celu musimy zmodyfikować syntaks funkcji **coxph()**, użytej w punkcie 3, poprzez dołączenie zmiennej **mutation** do modelu. Odpowiednie polecenie to:

```
nsclc.PH.P53em <- coxph(Surv(survtime, survind) ~ mutation + expression, data=nsclc)
```

Obiekt **nsclc.PH.P53em** zawiera szczegóły oszacowanego modelu. Syntax **Surv(survtime, survind) ~ mutation + expression** wskazuje, że czas przeżycia jest modelowany przy użyciu dwóch zmiennych niezależnych, **mutation** i **expression**. Rezultaty modelu sprawdzamy przy użyciu polecenia:

```
print(nsclc.PH.P53em)
```

5. Ocenimy założenia modelu PH z ekspresją białka P53 dla danych NSCLC. Wykres transformacji log-log krzywych przeżycia uzyskujemy przy użyciu metody Kaplana-Meiera uzyskujemy przy użyciu funkcji **plot()** w następującej postaci:

```
plot(nsclc.KM.P53e, col=c("red","blue"),fun=function(x) log(-log(x)),
log="x", firstx=1)
```

Argument **col=c("red","blue")** definiuje kolory krzywych przeżycia, a **fun=function(x) log(-log(x))** definiuje funkcję, której używamy do transformacji krzywych. **log="x"** wskazuje, że oś odciętych ma być przekształcona przy pomocy logarytmu, a **firstx=1** definiuje minimalną wartość na osi odciętych, która ma być oznaczana na wykresie. Otrzymany wykres odpowiada rycinie przedstawionej na wykładzie; różnica wynika użycia przez R krzywej „schodkowej". Oszacowania logarytmu funkcji skumulowanego hazardu nie odbiegaja dramatycznie od równoległych.

Formalny test założenia PH dla ekspresji białka możemy przeprowadzić przy użyciu testu Schoenfelda. W tym celu użyjemy funkcji **cox.zph()**:

```
nsclc.PHfit.P53e <- cox.zph(nsclc.PH.P53e, transform="identity")
```

Funkcja używa rezultatów oszacowania modelu PH z ekspresją białka P53 jako jedyną zmienną niezależną, które przechowywane są w obiekcie **nsclc.PH.P53e** (zob. punkt 3). W teście używamy oryginalnych wartości czasu, co wskazuje argument **transform="identity"**. Aby zastosować przekształcenie logarytmiczne czasu, należałoby podać wartość argumentu **transform="log"**. Wynik testu sprawdzamy poprzez wydrukowanie obiektu **nsclc.PHfit.P53e**, w którym zachowaliśmy ten wynik:

```
print(nsclc.PHfit.P53e)
```

Otrzymana wartość krytyczna testu *p* odpowiada wartości podanej na wykładzie.

Wskazane jest sprawdzenie wykresu skalowanych reszt Schoenfelda, odpowiadającego testowi. W tym celu używamy funkcji **plot()** w odniesieniu do obiektu **nsclc.PHfit.P53e**:

```
plot(nsclc.PHfit.P53e, df=4, nsmo=10, se=TRUE)
```

Aby dodać horyzontalną linię y=0.786 (gdzie 0.786 jest oszacowaniem współczynnika dla ekspresji białka, który znajduje się w obiekcie **nsclc.PH.P53e$coeff[1]**) do wykresu, należy – bez zamykania okna grafiki – wykonać polecenie:

```
abline(nsclc.PH.P53e$coeff[1], 0, lty=3)
```

Dodaje ono prostą o równaniu y=0.786 + 0x do aktywnego wykresu; **lty=3** wskazuje rodzaj linii (kropkowany) jaki chcemy uzyskać.

Otrzymujemy wykres odpowiadający rycinie przedstawionej na wykładzie. Różnica dotyczy użytej metody wygładzania: na rycinie przedstawionej na wykładzie użyto metody loess, a w poleceniu **plot()** zastosowano kubiczny spline (opcja **df=4**) dla 10 punktów (opcja **nsmo=10**). Linowy lub kwadatrowy spline wymagałby użycia opcji, odpowiednio, **df=2** i **df=3**. Wygładzona krzywa odbiega od linii prostej, ale należy wziąć pod uwagę efekty „brzegów". Dołączone granice dla 95% obszaru ufności nie dają podstawy do odrzucenia hipotezy, że krzywa różni się od horyzontalnej.

Jeśli niepokoi nas nie-liniowość, możemy użyć np. kwadratowej funkcji czasu. W tym celu stosujemy następujące polecenia:

```
nsclc.PHfit1.P53e <- cox.zph(nsclc.PH.P53e, transform=function(x) x^2)
plot(nsclc.PHfit1.P53e, df=4, nsmo=10, se=TRUE)
abline(0.786, 0, lty=3)
```

Wykres i wynik testu nie sugerują odstępstw od założenia PH.

6. Ocenimy założenia modelu PH z ekspresją białka P53 i mutacją genu P53 dla danych NSCLC.

Ogólne dopasowanie modelu sprawdzimy przy użyciu reszt opartych na dewiancji. Uzyskujemy je z obiektu **nsclc.PH.P53em** przy użyciu funkcji **residuals()** z argumentem **type="deviance"**:

```
nsclc.devres.P53em <- residuals(nsclc.PH.P53em,type="deviance")
```

Potrzebujemy również wartości linowej kombinacji zmiennych niezależnych. Uzyskujemy je z obiektu **nsclc.PH.P53em** przy użyciu funkcji **predict()** z argumentem **type="lp"** (linear predictor):

```
nsclc.fitval.P53em <- predict(nsclc.PH.P53em,type="lp")
```

Następnie tworzymy wykres reszt w funkcji wartości kombinacji zmiennych niezależnych przy użyciu funkcji **plot()**:

```
plot(nsclc.fitval.P53em,nsclc.devres.P53em)
```

Wykres odpowiada rycinie przedstawionej na wykładzie – różnica w wartościach na osi odciętych wynika z faktu centrowania zmiennych niezależnych przez R. Wykres wskazuje na możliwość niedopasowania modelu.

W celu sprawdzenia założenia PH dla mutacji genu P53, wyznaczamy krzywe przeżycia metodą Kaplana-Meiera:

```
nsclc.KM.P53m <- survfit(Surv(survtime, survind) ~ mutation, data=nsclc)
```

Oszacowane krzywe przeżycia przedstawiamy graficznie przy pomocy funkcji **plot()**:

```
plot(nsclc.KM.P53m, col=c("red","blue"), xlab="months",ylab="survival
probability")
```

Otrzymane krzywe przecinają się. Wykres transformacji log-log krzywych uzyskujemy poprzez użycie dodatkowych argumentów funkcji **plot()**:

```
plot(nsclc.KM.P53m, col=c("red","blue"),fun=function(x) log(-log(x)),
log="x", firstx=1)
```

Otrzymany wykres odpowiada rycinie przedstawionej na wykładzie; różnica wynika użycia przez R krzywej „schodkowej". Oczywiście, krzywe przecinają się, sugerując niespełnianie od założenia PH.

Formalny test założenia PH przeprowadzamy przy użyciu testu Schoenfelda:

```
nsclc.PHfit.P53em <- cox.zph(nsclc.PH.P53em, transform="identity")
```

Funkcja **cox.zph()** używa odpowiednich rezultatów przechowywanych w obiekcie **nsclc.PH.P53em**. W teście używamy oryginalnych wartości czasu. Wynik testu sprawdzamy poprzez wydrukowanie obiektu **nsclc.PHfit.P53em**:

```
print(nsclc.PHfit.P53em)
```

Otrzymujemy rezultaty zgodne z wynikami podanym na wykładzie. Założenie PH nie jest spełnione dla mutacji genu P53.

Wykresy skalowanych reszt Schoenfelda, odpowiadające testowi, uzyskujemy przy pomocy funkcji **plot()** w odniesieniu do obiektu **nsclc.PHfit.P53em**:

```
plot(nsclc.PHfit.P53em, df=4, nsmo=10, se=TRUE, var=1)
abline(nsclc.PH.P53em$coeff[1], 0, lty=3)
plot(nsclc.PHfit.P53em, df=4, nsmo=10, se=TRUE, var=2)
abline(nsclc.PH.P53em$coeff[2], 0, lty=3)
```

Wykres dla mutacji (**var=1**) sugeruje nielinowość trendu i możliwość uzycia funkcji logarytmicznej w konstrukcji testu. Krzywa dla ekspresji białka (**var=2**) w głównej części jest liniowa (pomijając „brzegowe" efekty po lewej stronie).

7. Rozważmy model PH z ekspresją białka P53, warstwowany ze względu na mutację genu P53.

W tym celu musimy zmodyfikować syntaks funkcji **coxph()**, użytej w punkcie 6, i wskazać zmienną **mutation** jako warstwującą. Odpowiednie polecenie to:

```
nsclc.strPH.P53em <- coxph(Surv(survtime, survind) ~ expression +
strata(mutation), data=nsclc)
```

Wyniki modelu zapisujemy w obiekcie **nsclc.strPH.P53em**, który następnie drukujemy::

```
print(nsclc.strPH.P53em)
```

Rezultaty odpowiadają wynikom przedstawionym na slajdzie 104.

Ogólne dopasowanie modelu sprawdzamy przy użyciu reszt opartych na dewiancji. Uzyskujemy je z obiektu **nsclc.strPH.P53em** przy użyciu funkcji **residuals()** z argumentem **type="deviance"**:

```
nsclc.devres1.P53em <- residuals(nsclc.strPH.P53em,type="deviance")
```

Wartości linowej kombinacji zmiennych niezależnych możemy otrzymać przez odwołanie się do składnika **linear.predictors** obiektu **nsclc.strPH.P53em**:

```
nsclc.fitval1.P53em <- nsclc.strPH.P53em$linear.predictors
```

Wykres reszt w funkcji wartości kombinacji zmiennych niezależnych otrzymujemy przy użyciu funkcji **`plot()`**:

```
plot(nsclc.fitval1.P53em,nsclc.devres1.P53em)
```

Wykres odpowiada rycinie przedstawionej na wykładzie – różnica w wartościach na osi odciętych wynika z faktu centrowania zmiennych niezależnych przez R.

7. Pakiet *survival* zawiera dane dotyczące przeżycia chorych na raka jajnika. Jedną ze zmiennych objaśniających jest wiek chorych. By użyć wieku jako zmiennej ciągłej w modelu PH powinniśmy użyc odpowiedniej funkcji wieku. W celu wyboru funkcji posłużymy się wykresem opartym na resztach martyngałowych.

Najpierw dopasowujemy „pusty" model PH:

```
ovar.PH <-  coxph(Surv(futime,fustat)~1, data=ovarian)
```

Z obiektu wynikowego „wyciągamy" reszty martyngałowe (domyślne):

```
mart <- resid(ovar.PH)
```

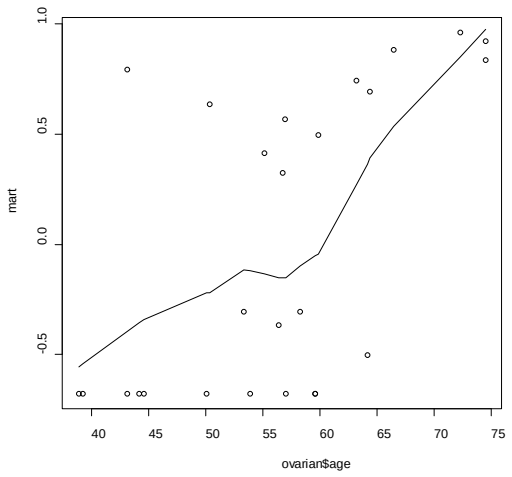
Następnie konstruujemy wykres reszt względem wieku:

```
plot(ovarian$ge,mart)
```

I dodajemy do niego gładką krzywą uzyskaną przy użyciu metody lowess:

```
lines(lowess(ovarian$age,mart,iter=0,f=0.6))
```

Dla każdej obserwacji, lowess używa „lokalnej" (opartej na f*n sąsiednich punktach, gdzie f jest wskazywane argumentem **`f`**) ważonej regresji liniowej do wyznaczenia „oczekiwanej" wartości obserwacji. Argument **`iter=0`** jest ważny, bo zapobiega iteracyjnemu „poprawianiu" szacowania krzywej poprzez usuwanie oberwacji odstających (może się to skończyć np. usunięciem wszystkich reszt odpowiadających zdarzeniom...). W efekcie otrzymujemy następujący wykres sugerujący możliwość użycia funkcji liniowej:



Sesja komputerowa 2: SAS

1. Dane dotyczące NSCLC znajdują się w pliku `nsclc_eng.sas7bdat`. Aby uzyskać do nich dostęp, musimy najpierw wskazać katalog, w którym znajduje się plik. W tym celu używamy polecenia `libname`:

```
libname pw " t:/burzykowskit/biostat/datasets";
```

2. Podstawową procedurą dla potrzeb szacowania modelu PH jest PROC PHREG.

Aby uzyskać oszacowanie modelu z ekspresją białka P53, używamy następującego syntaksu:

```
proc phreg data=pw.nsclc_eng;
      model survtime*survind(0) = expression / ties=efron;
run;
```

Polecenie `model survtime*survind(0) = expression;` wskazuje zmienną zawierającą czas obserwacji i wskaźnik cenzurowania. W nawiasie podawane są wartości, które identyfikują obserwacje cenzurowane. Ponadto polecenie identyfikuje zmienną `expression` jako zmienną niezależną, która ma się znaleźć w modelu. Opcja `ties=efron` używa metody Efrona do obliczenia częściowej funkcji wiarogodności. Należy pamiętać, że domyślnie używana jest mniej dokładna metoda Breslowa. W naszym przypadku nie ma to znaczenia, bowiem dane nie zawierają jednoczesnych zdarzeń.

Wyniki pojawiają się w oknie *Output*. Odpowiadają wynikom przedstawionym na wykładzie. Jako że model zawiera tylko jedną zmienną, podane w wydruku wyniki testów ilorazu wiarogodności, „score", i Walda dla globalnej hipotezy zerowej, że wszystkie współczynniki modelu wynoszą 0, są w efekcie wynikami tylko dla ekspresji białka.

Aby uzyskać przedziały ufności dla ilorazu hazardu, musimy użyć opcji `rl`:

```
proc phreg data=pw.nsclc_eng;
      model survtime*survind(0) = expression / ties=efron rl;
run;
```

Aby uzyskać oszacowania funkcji przeżycia oparte na modelu, musimy najpierw utworzyć zbiór danych z wartościami zmiennej niezależniej, dla których chcemy uzyskać oszacowania:

```
data predcov;
 input expression;
 datalines;
 0
 1
;
 run;
```

Następnie dopasowujemy model z użyciem polecenia `baseline`:

```
proc phreg data= pw.nsclc_eng ;
 model survtime*survind(0) = expression / ties=efron;
```

```
 baseline covariates=predcov out=PredS survival=S cumhaz=ch / nomean;
run;
```

Użyta forma polecenia `baseline` wskazuje, że oszacowania funkcji przeżycia mają być wyznaczone dla wartości zmiennych niezależnych zawartych w pliku `predcov`. Oszacowania mają być zapisane w zbiorze `PredS`. Składniki `survival=S loglogs=llS` wskazują, że interesują nas oszacowania funkcji przeżycia, które powinny być zachowane jako zmienna `S`, oraz funkcji skumulowanego hazardu, które powinny być zachowane jako zmienna `ch`. Opcja `nomean` wyklucza z rezultatów wartość oszacowań dla średniej próbkowej zmiennej niezależnej ze zbioru użytego do szacowania modelu.

Uzyskane oszacowania dla funkcji przeżycia możemy wykreślić przy pomocy procedury PROC GPLOT:

```
   proc gplot data=PredS;
      plot S*survtime=expression;
      symbol1 interpol=stepLJ c=blue;
      symbol2 interpol=stepLJ c=red;
   run;
 quit;
```

Podobny syntaks może zostać użyty do wykreślenia oszacowań funkcji sumulowanego hazardu:

```
   proc gplot data=PredS;
      plot ch*survtime=expression;
      symbol1 interpol=stepLJ c=blue;
      symbol2 interpol=stepLJ c=red;
   run;
 quit;
```

Wykres odpowiada wykresowi przedstawionemu na wykładzie.

3. Ocenimy założenia modelu PH z ekspresją białka P53 dla danych NSCLC.

Wykres transformacji log-log krzywych przeżycia, uzyskanych metodą Kaplana-Meiera, uzyskujemy poprzez użycie PROC LIFETEST z opcją `plots=(lls)`:

```
proc lifetest data= pw.nsclc_eng plots=(lls);
 time survtime*survind(0) ;
 strata expression;
 run;
```

Otrzymany wykres odpowiada rycinie przedstawionej na wykładzie.

W PROC PHREG nie ma implementacji testu Schonfelda. Ale możemy w łatwy sposób użyć testu opartego na współczynniku zależnym od czasu. W tym celu używamy następującego syntaksu:

```
proc phreg data= pw.nsclc_eng ;
 exp_t=expression*survtime;
 model survtime*survind(0) = expression exp_t / ties=efron;
run;
```

Głównym krokiem jest utworzenie zmiennej `exp_t=expression*survtime`, będącej interakcją wskaźnika ekspresji białka P53 i czasu. Zmienna ta jest następnie użyta jako dodatkowa zmienna niezależna w poleceniu `model survtime*survind(0) = expression exp_t`. Wyniki w oknie *Output* zawierają test Walda dla tej zmiennej. Rezultaty zgadzają się z wynikami przedstawionymi na wykładzie.

PROC PHREG daje możliwość obliczania różnego rodzaju reszt. Uzyskuje się je przy pomocy odpowiedniej formy polecenia `output`. Na przykład, w celu uzyskania skalowanych reszt Schoenfelda, używamy następującego syntaksu:

```
proc phreg data= pw.nsclc_eng ;
 model survtime*survind(0) = expression / ties=efron;
 output out=resids wtressch=ssch;
run;
```

Użyta forma polecenia `out` wskazuje, że reszty mają być zapisane w zbiorze `resids`. W szczególności, interesują nas skalowane reszty Schoenfelda, które w SASie nazywane są „ważonymi" resztami Schoenfelda, które powinny być zachowane jako zmienna `ssch`.

Wykres reszt uzyskujemy przy pomocy procedury PROC GPLOT:

```
   proc gplot data=resids;
      plot ssch*survtime;
      symbol1 interpol=sm70s value=circle c=blue;
   run;
 quit;
```

W szczególności, w graficznej opcji `symbol1` używamy składnika `interpol=sm70s`, który dodaje do wykresy gładką krzywą (`interpol=sm`) opartą na „wygładzaniu" 70% danych (`interpol=sm70`). Przed wygładzaniem dane są sortowane (`interpol=sm70s`). Otrzymujemy wykres odpowiadający rycinie przedstawionej na slajdzie 98. Różnica dotyczy użytej metody wygładzania.

4. Rozważmy model PH z ekspresją białka P53 i mutacją genu P53. Jego oszacowanie uzyskujemy przy pomocy następującego syntaksu:

```
proc phreg data= pw.nsclc_eng ;
  model survtime*survind(0) = expression mutation / ties=efron rl;
run;
```

Uzyskane wyniki odpowiadają rezultatom pokazanym na wykładzie.

Domyślnie, PROC PHREG raportuje dla poszczególnych współczynników modelu jedynie wyniki testu Walda. Można jednak uzyskać również rezultat testu „score". W tym celu trzeba jednak skorzystać z algorytmu „automatycznego" wyboru zmiennych. Na przykład, dla zmiennej `mutation` możemy użyć następującego syntaksu:

```
proc phreg data= pw.nsclc_eng ;
      model survtime*survind(0) = expression mutation / ties=efron
                        selection=forward start=1 sle=1;
run;
```

Opcja `selection=forward` wskazuje, że modele mają być budowane poprzez dodawanie kolejnych zmiennych. Opcja `start=1` oznacza, że zaczynamy nie od pustego modelu, ale od modelu zawierającego pierwszą zmienną podaną w poleceniu `model`, tj. `expression.` Opcja `sle=1` oznacza, że do modelu dołączamy tę z pozostałych zmiennych wymienionych w poleceniu `model`, dla której poziom krytyczny p testu „score", uwzględniającego zmienne już włączone do modelu (tj. `expression`) jest najmniejszy i mniejszy niż 1. Innymi słowy, niezależnie od wyniku testu, któraś ze zmiennych zostanie dołączona. W naszym przypadku, będzie to zmienna `mutation`.

Wyniki przedstawione w oknie *Output* zawierają informację (`Summary of Forward Selection`), że wartość statystyki testu „score" dla zmiennej `mutation` wynosi 18.4967, a $p<0.0001$. Rezultat ten jest bliski wynikowi testu Walda (18.1662 z $p<0.0001$).

W podobny sposób możemy uzyskać wynik testu „score" dla zmiennej `expression`. Musimy jedynie zmienić kolejność zmiennych w poleceniu `model`:

```
proc phreg data= pw.nsclc_eng ;
      model survtime*survind(0) = mutation expression / ties=efron
                        selection=forward start=1 sle=1;
run;
```

Wyniki przedstawione w oknie *Output* zawierają informację (`Summary of Forward Selection`), że wartość statystyki testu „score" dla zmiennej `expression` wynosi 0.2893, a $p=0.5907$. Rezultat ten jest bliski wynikowi testu Walda (0.2898 z $p=0.5904$).

PROC PHREG nie daje możliwości automatycznego uzyskania wyników testu ilorazu wiarogodności dla poszczególnych współczynników modelu. Testy te można uzyskać jedynie „ręcznie" poprzez szacowanie modelu z i bez interesującego nas współczynnika, a następnie obliczenie różnicy logarytmów częściowych funkcji wiarogodności dla uzyskanych modeli.

5. Ocenimy założenia PH dla zmiennej `mutation` w modelu z ekspresją białka P53 i mutacją genu P53 dla danych NSCLC. W tym celu użyjemy testu Walda dla współczynnika zależnego od czasu:

```
proc phreg data= pw.nsclc_eng ;
  mut_t=mutation*survtime ;
  model survtime*survind(0) = expression mutation mut_t / ties=efron rl;
run;
```

Wyniki odpowiadają rezultatom przedstawionym na wykładzie.

6. Rozważmy model PH z ekspresją białka P53, warstwowany ze względu na mutację genu P53. Jego oszacowanie uzyskujemy przy pomocy następującego syntaksu:

```
proc phreg data= pw.nsclc_eng ;
  model survtime*survind(0) = expression / ties=efron;
  strata mutation ;
run;
```

Polecenie `strata` pozwala nam na wskazanie zmiennej `mutation` jako definiującej warstwy. Uzyskane wyniki odpowiadają rezultatom podanym na wykładzie.

Wykres oceniający ogólne dopasowanie modelu konstruujemy przy użyciu reszt opartych na dewiancji w następujący sposób

```
proc phreg data= pw.nsclc_eng ;
 model survtime*survind(0) = expression / ties=efron;
 strata mutation ;
 output out=resids1 resdev=dev ;
run;

   proc gplot data=resids1;
      plot dev*expression / haxis=-.05 to 1.05 by .05;
      symbol1 interpol=none value=circle c=blue;
   run;
 quit;
```

7. Dane dotyczące przeżycia chorych na raka jajnika znajdują się w pliku **`ovarian.sas7bdat`**. Użyjemy wykresu reszt martyngałowych do wyboru formy funkcjonalnej wieku., która powinna być zastosowana do zmiennej.

Najpierw musimy dopasować „pusty" model PH. W tym celu używamy sztuczki – definiujemy zmienną `dummy` równą 1 dla wszystkich obserwacji i używamy jej w PROC PHREG:

```
data ovarian;
  set pw.ovarian;
  dummy=1;
 run;

proc phreg data= ovarian ;
 model futime*fustat(0) = dummy / ties=efron;
 id age ;
 output out=resids2 resmart=mart ;
run;
```

Użyta forma polecenia `out` wskazuje, że reszty mają być zapisane w pliku `resids2` i że interesują nas skalowane reszty martyngałowe, które powinny być zachowane jako zmienna `mart`. Polecenie `id age ;` "dokłada" zmienną `age` do pliku `resids2`.

Następnie sortujemy `resids2` ze względu na zmienną `age`:

```
  proc sort data=resids2 ;
    by age ;
  run;
```

i używamy PROC LOESS do obliczenia gładkiej krzywej dla reszt w funkcji zmiennej `age`:

```
  proc loess data=resids2;
     model mart = age / smooth=0.6;
     ods output OutputStatistics=smoothed;
  run ;
```

Opcja `smooth=0.6` definiuje, że dla każdej obserwacji „sąsiedztwo” składa się z 60% wszystkich obserwacji. Wyniki są zapisane w pliku `smoothed`. Plik ten sortujemy ze względu na zmienną `age`:

```
proc sort data=smoothed ;
   by age ;
run ;
```

i używamy PROC GPLOT do uzyskania odpowiedniego wykresu względem wieku:

```
proc gplot data=smoothed;
  plot  DepVar*age Pred*age/ overlay;
  symbol2 interpol=l;
run;
quit ;
```

W szczególności, polecenie `plot  DepVar*age Pred*age/ overlay;` definiuje dwa wykresy: jeden dla zmiennej zależnej `DepVar` (tj. reszt) a drugi dla zmiennej `Pred`, która zawiera wartości dla gładkiej krzywej; nazwy tych zmiennych są domyślne w PROC LOESS. Opcja `overlay` „nakłada” na siebie te dwa wykresy. Opcja graficzna `symbol2 interpol=l;` wskazuje, że w drugim z tych wykresów punkty mają być połączone linią. Uzyskany wykres odpowiada wykresowi skonstruowanemu w punkcie 7 części R i sugeruje możliwość użycia funkcji liniowej.