R 使い方入門：重回帰分析・ダミー変数・頑健な標準誤差

重回帰分析

データ読み込み

このチャンクでは、`read.csv()` 関数を使用して `6_1_income.csv` ファイルを読み込み、データフレーム `df` に格納します。`head()` 関数で最初の数行を表示することで、`lincome`（対数賃金）、`yeduc`（教育年数）、`exper`（経験年数）といった変数が正しく読み込まれているかを確認します。これは、分析を始める前の基本的なデータ検証ステップです。

```
df <- read.csv("6_1_income.csv")
head(df)
#>   exper exper2 yeduc income  lincome
#> 1     7     49     9    100 4.605170
#> 2     8     64     9    150 5.010635
#> 3     8     64     9    150 5.010635
#> 4    10    100     9    200 5.298317
#> 5    10    100     9    300 5.703783
#> 6    11    121     9    150 5.010635
```

主な変数は以下の通りです。

- `lincome`: 賃金の対数値
- `yeduc`: 教育年数
- `exper`: 経験年数
- `exper2`: 経験年数の二乗

重回帰分析

ここでは、`lm()` 関数を用いて、ミンサー方程式に基づいた重回帰モデルを推定します。対数賃金 `lincome` を目的変数とし、教育年数 `yeduc`、経験年数 `exper`、およびその二乗 `exper2` を説明変数とします。`summary()` で表示される結果（今回はテキストに記述）を通じて、各変数の係数、p 値、モデル全体の決定係数などを評価します。

```r
model1 <- lm(lincome ~ yeduc + exper + exper2, data = df)
summary(model1)
#>
#> Call:
#> lm(formula = lincome ~ yeduc + exper + exper2, data = df)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -4.0389 -0.3214  0.1681  0.5124  2.1326
#>
#> Coefficients:
#>               Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  2.4855019  0.1107823   22.44   <2e-16 ***
#> yeduc        0.1175467  0.0070603   16.65   <2e-16 ***
#> exper        0.1961736  0.0074935   26.18   <2e-16 ***
#> exper2      -0.0063811  0.0003162  -20.18   <2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.7983 on 4295 degrees of freedom
#> Multiple R-squared:  0.2066, Adjusted R-squared:  0.206
#> F-statistic: 372.8 on 3 and 4295 DF,  p-value: < 2.2e-16
```

`summary(model1)` の結果、`yeduc`、`exper`、`exper2` のすべての係数が 1% 水準で統計的に有意であることがわかります。

- **yeduc**: 教育年数が 1 年増えると、収入が平均して約 11.8% 増加することを示します（係数: 0.118)。
- **exper, exper2**: 経験年数は収入に対して正の効果（係数: 0.196）を持ちますが、その効果は経験を積むごとに減少していきます（`exper2` の係数が負: -0.006)。
- **Adjusted R-squared**: 調整済み決定係数は 0.206 であり、このモデルが対数収入のばらつきの約 20.6% を説明することを示します。

回帰解剖

FWL（Frisch-Waugh-Lovell）定理を利用して「回帰解剖」を行います。まず、`yeduc` を他の説明変数（`exper` と `exper2`）に回帰させ、その残差を `model10` から抽出します。次に、`lincome` をこの残差に回帰させます（`model11`)。この単回帰の係数が、元の重回帰モデル `model1` における `yeduc` の係数と一致することを確認します。

```r
model10 <- lm(yeduc ~ exper + exper2, data = df)
model11 <- lm(lincome ~ residuals(model10), data = df)
```

```
summary(model11)
#>
#> Call:
#> lm(formula = lincome ~ residuals(model10), data = df)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -3.7719 -0.3036  0.2205  0.5544  2.0821
#>
#> Coefficients:
#>                    Estimate Std. Error t value Pr(>|t|)
#> (Intercept)        5.290452   0.013311  397.45   <2e-16 ***
#> residuals(model10) 0.117547   0.007719   15.23   <2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.8727 on 4297 degrees of freedom
#> Multiple R-squared:  0.0512, Adjusted R-squared:  0.05098
#> F-statistic: 231.9 on 1 and 4297 DF,  p-value: < 2.2e-16
```

`model11` における残差項の係数は 0.1175 であり、これは `model1` の `yeduc` の係数と完全に一致します。これは FWL 定理を実証しており、教育年数が賃金に与える効果が、経験年数の影響を統制した上での純粋な効果として正しく推定されていることを示しています。

対数変換

`log()` 関数を用いて `income` 変数を対数変換し、`lincome` と同様のモデルを推定します。`lincome` が `log(income)` の対数変換であることを確認し、対数線形モデルでは係数が「賃金のパーセント変化」として解釈できることを示します。

```
model2 <- lm(log(income) ~ yeduc + exper + exper2, data = df)
summary(model2)
#>
#> Call:
#> lm(formula = log(income) ~ yeduc + exper + exper2, data = df)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -4.0389 -0.3214  0.1681  0.5124  2.1326
#>
#> Coefficients:
```

```
#>                 Estimate Std. Error t value Pr(>|t|)    
#> (Intercept)  2.4855022  0.1107823   22.44   <2e-16 ***
#> yeduc        0.1175467  0.0070603   16.65   <2e-16 ***
#> exper        0.1961736  0.0074935   26.18   <2e-16 ***
#> exper2      -0.0063811  0.0003162  -20.18   <2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#> 
#> Residual standard error: 0.7983 on 4295 degrees of freedom
#> Multiple R-squared:  0.2066, Adjusted R-squared:  0.206 
#> F-statistic: 372.8 on 3 and 4295 DF,  p-value: < 2.2e-16
```

`model2` の推定結果は `model1` と完全に一致します。これは、データフレームの `lincome` 列がもともと `income` の自然対数として計算されていたためです。このチャンクは、`log(income)` と `lincome` が同じ結果をもたらすことを確認し、分析の一貫性を示しています。

二乗項

`lm()` の式の中で `I(exper^2)` を使用し、事前に二乗項の変数を作成することなく、経験年数の非線形な効果をモデルに組み込みます。この `model3` が、`exper2` を事前に作成した `model1` と同一の結果になることを確認します。

```
model3 <- lm(lincome ~ yeduc + exper + I(exper^2), data = df)
summary(model3)
#> 
#> Call:
#> lm(formula = lincome ~ yeduc + exper + I(exper^2), data = df)
#> 
#> Residuals:
#>     Min      1Q  Median      3Q     Max 
#> -4.0389 -0.3214  0.1681  0.5124  2.1326 
#> 
#> Coefficients:
#>                 Estimate Std. Error t value Pr(>|t|)    
#> (Intercept)  2.4855019  0.1107823   22.44   <2e-16 ***
#> yeduc        0.1175467  0.0070603   16.65   <2e-16 ***
#> exper        0.1961736  0.0074935   26.18   <2e-16 ***
#> I(exper^2)  -0.0063811  0.0003162  -20.18   <2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#> 
```

```
#> Residual standard error: 0.7983 on 4295 degrees of freedom
#> Multiple R-squared:  0.2066, Adjusted R-squared:  0.206
#> F-statistic: 372.8 on 3 and 4295 DF,  p-value: < 2.2e-16
```

model3 の結果は model1 と完全に一致します。これにより、lm() の式の中で I() 関数を使って変数を変換する手法が、事前に新しい変数を作成する手法と等価であることが実証されます。

F 検定

exper と exper2 をモデルに加えることの統計的有意性を F 検定で評価します。yeduc のみを含む制約モデル model0 と、exper と exper2 を追加した非制約モデル model1 を比較します。残差平方和（SSR）を用いて手計算で F 統計量を算出し、検定の仕組みを理解します。

```
model0 <- lm(lincome ~ yeduc, data = df)
anova(model0, model3)
#> Analysis of Variance Table
#>
#> Model 1: lincome ~ yeduc
#> Model 2: lincome ~ yeduc + exper + I(exper^2)
#>   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
#> 1   4297 3373.8
#> 2   4295 2736.9  2    636.92 499.75 < 2.2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

F 統計量は 499.75、p 値はほぼゼロとなります。これは、exper と exper2 の係数が両方ともゼロであるという帰無仮説を強く棄却することを示します。したがって、経験年数とその二乗項をモデルに加えることは、モデルの当てはまりを統計的に有意に改善すると結論付けられます。手計算による F 値も anova の結果と一致し、5% 水準の F 臨界値 2.99 を大幅に上回ります。

```
# 残差二乗和の計算
ssr1 <- deviance(model3)
ssr0 <- deviance(model0)
# 分子 (numerator)
k <- model3$rank - model0$rank #  説明変数の違い
f0 <- (ssr0 - ssr1) / k
# 分母 (denominator)
dof <- model3$df.residual # 自由度 (degree of freedom)
f1 <- ssr1 / dof
# F 統計量の計算
fstat <- f0 / f1
```

```
fstat
#> [1] 499.7542
```

ダミー変数

0 もしくは 1 をとる変数をダミー変数といいます。R では、カテゴリ変数を自動的にダミー変数に変換して回帰分析を行うことができます。カテゴリ変数は因子（factor）として扱われ、`lm()` 関数は自動的にダミー変数を生成します。2 種類だけでなく 3 種類のカテゴリも扱えますが、ここでは省略します。このセクションでは、性別ダミー変数を用いて男女間の賃金格差と教育の収益率の違いを分析します。

定数項のみ

`7_1_income.csv` から性別ダミー変数 `female` を含む新しいデータを読み込み、`yeduc` と `female` を説明変数とするモデルを推定します。`female` の係数が、教育年数が同じ場合に男女間の賃金格差をどの程度示すかを確認します。

```
df <- read.csv("7_1_income.csv")
model1 <- lm(lincome ~ yeduc + female, data = df)
summary(model1)
#>
#> Call:
#> lm(formula = lincome ~ yeduc + female, data = df)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -4.0269 -0.2681  0.1906  0.5091  2.2592
#>
#> Coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  4.863272   0.096125  50.593   <2e-16 ***
#> yeduc        0.058598   0.006739   8.696   <2e-16 ***
#> female      -0.832148   0.025349 -32.827   <2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.8268 on 4283 degrees of freedom
#> Multiple R-squared:  0.2203, Adjusted R-squared:  0.2199
#> F-statistic:   605 on 2 and 4283 DF,  p-value: < 2.2e-16
```

`summary(model1)` の結果、`female` の係数は-0.832 と推定され、統計的にも非常に有意です。これは、教育

年数を同じとした場合、女性は男性に比べて平均的に対数賃金が 0.832 低い（賃金が約 56% 低い）ことを示唆しており、顕著な男女間の賃金格差が存在することを示しています。

交差項の導入

教育の収益率（`yeduc` の係数）が性別によって異なる可能性を検証するため、`female` と `yeduc` の交差項 `female_yeduc` をモデルに加えます。この交差項の係数は、女性であることによって教育の収益率がどれだけ変化するかを示します。

```
model2 <- lm(lincome ~ yeduc + female + female_yeduc, data = df)
summary(model2)
#>
#> Call:
#> lm(formula = lincome ~ yeduc + female + female_yeduc, data = df)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -3.9239 -0.2630  0.1978  0.4854  2.2936
#>
#> Coefficients:
#>               Estimate Std. Error t value Pr(>|t|)
#> (Intercept)   5.346896   0.120920  44.218  < 2e-16 ***
#> yeduc         0.024095   0.008533   2.824  0.00477 **
#> female       -2.079202   0.192386 -10.807  < 2e-16 ***
#> female_yeduc  0.090229   0.013800   6.538 6.94e-11 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.8228 on 4282 degrees of freedom
#> Multiple R-squared:  0.228,  Adjusted R-squared:  0.2274
#> F-statistic: 421.5 on 3 and 4282 DF,  p-value: < 2.2e-16
```

`summary(model2)` の結果、交差項 `female_yeduc` の係数は 0.090 と正で統計的に有意です。これは、教育の収益率が男女で異なることを示唆しています。

- **男性（基準）の教育収益率**: `yeduc` の係数である 0.024。
- **女性の教育収益率**: $0.024 + 0.090 = 0.114$。

この結果は、女性は平均賃金が低いものの、教育年数が 1 年増えることによる賃金の上昇率は男性よりも高いことを意味します。

交差項の別表記

R の formula 機能である : 演算子を用いて、`female` と `yeduc` の交差項をモデルに投入します。`female:yeduc` という記述は、事前に `female_yeduc` 変数を手動で作成するのと同じ効果を持ちます。

```
model3 <- lm(lincome ~ yeduc + female + female:yeduc, data = df)
summary(model3)
#>
#> Call:
#> lm(formula = lincome ~ yeduc + female + female:yeduc, data = df)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -3.9239 -0.2630  0.1978  0.4854  2.2936
#>
#> Coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  5.346896   0.120920  44.218  < 2e-16 ***
#> yeduc        0.024095   0.008533   2.824  0.00477 **
#> female      -2.079202   0.192386 -10.807  < 2e-16 ***
#> yeduc:female  0.090229   0.013800   6.538 6.94e-11 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.8228 on 4282 degrees of freedom
#> Multiple R-squared:  0.228,  Adjusted R-squared:  0.2274
#> F-statistic: 421.5 on 3 and 4282 DF,  p-value: < 2.2e-16
```

`model3` の結果は `model2` と完全に一致します。これは、`:` 演算子を使って交差項を指定する方法が、手動で交差項の変数を作成する方法と等価であることを確認できます。

交差項の別表記

`*`演算子は、`yeduc` と `female` の主効果とそれらの交差項 `yeduc:female` を一度にモデルに含めるための便利な省略記法です。`lm(lincome ~ yeduc * female, ...)` は `lm(lincome ~ yeduc + female + yeduc:female, ...)` と等価です。

```
model4 <- lm(lincome ~ yeduc * female, data = df)
summary(model4)
#>
#> Call:
```

```
#> lm(formula = lincome ~ yeduc * female, data = df)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -3.9239 -0.2630  0.1978  0.4854  2.2936
#>
#> Coefficients:
#>               Estimate Std. Error t value Pr(>|t|)
#> (Intercept)   5.346896   0.120920  44.218  < 2e-16 ***
#> yeduc         0.024095   0.008533   2.824  0.00477 **
#> female       -2.079202   0.192386 -10.807  < 2e-16 ***
#> yeduc:female  0.090229   0.013800   6.538 6.94e-11 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.8228 on 4282 degrees of freedom
#> Multiple R-squared:  0.228,  Adjusted R-squared:  0.2274
#> F-statistic: 421.5 on 3 and 4282 DF,  p-value: < 2.2e-16
```

`model4` の結果は `model2` および `model3` と同一です。これにより、`*`演算子が、複数の主効果とそれらの交差項を一度に指定するための有効なショートカットであることがわかります。

F 検定

性別ダミー `female` と交差項 `yeduc:female` を加えることで、モデルの当てはまりが統計的に有意に改善したかを F 検定で評価します。`yeduc` のみを含む制約モデル `model0` と、交差項を追加した非制約モデル `model4` を `anova()` 関数で比較します。

```
model0 <- lm(lincome ~ yeduc, data = df)
anova(model0, model4)
#> Analysis of Variance Table
#>
#> Model 1: lincome ~ yeduc
#> Model 2: lincome ~ yeduc * female
#>   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
#> 1   4284 3664.7
#> 2   4282 2899.1  2    765.65 565.45 < 2.2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

`anova(model0, model4)` による F 検定の結果、F 値は 565.45 と非常に大きく、p 値も極めて小さいです。

これは、性別ダミーと教育との交差項をモデルに加えることが、統計的に見て非常に有意義であり、モデルの説明力を大きく向上させることを示しています。

チャウ検定

チャウ検定は、サブグループ（ここでは性別）間で回帰係数が安定しているか（構造変化がないか）を検定します。model0（プールしたモデル）と model4（交差項を含むモデル）の残差平方和（SSR）を用いて、手計算で F 統計量を算出します。

```
# 男性のみのデータを使って単回帰で推定
reg1 <- lm(lincome ~ yeduc, data = subset(df, female == 0))
SSR1 <- deviance(reg1)
# 女性のみのデータを使って単回帰で推定
reg2 <- lm(lincome ~ yeduc, data = subset(df, female == 1))
SSR2 <- deviance(reg2)
# すべてのデータを使って単回帰で推定
reg3 <- lm(lincome ~ yeduc, data = df)
SSR <- deviance(reg3)
k <- reg3$rank
# チョウ検定統計量の計算
Fstat <- (SSR - (SSR1 + SSR2)) / (SSR1 + SSR2) * (nrow(df) - 2 * k) / k
Fstat
#> [1] 565.4484
```

手計算で求めた F 統計量（Fstat）は 565.45 となり、前の anova() の結果と一致します。この F 値は 5% 水準の臨界値 2.99 をはるかに超えているため、係数が男女間で等しいという帰無仮説は棄却されます。これは、男女間で賃金と教育の関係に「構造的な違い」があることの強い証拠となります。

頑健な標準誤差

不均一分散の問題に対処するため、頑健な標準誤差を計算します。lmtest と sandwich パッケージを使い、不均一分散に対して頑健な（HC1）標準誤差を計算し、係数の有意性を再評価します。

```
# データの読み込み
income6 <- read.csv("6_1_income.csv")

# ミンサー方程式を OLS で重回帰
reg1 <- lm(lincome ~ yeduc + exper + exper2, data = income6)

# 必要なパッケージのインストールと読み込み
library(lmtest)
```

```r
library(sandwich)

# 分散共分散行列を計算
vcov <- vcovHC(reg1, type = "HC1")
# 重回帰分析の結果と vcov から標準誤差をホワイト修正
coeftest(reg1, vcov)
#>
#> t test of coefficients:
#>
#>                 Estimate  Std. Error t value  Pr(>|t|)
#> (Intercept)  2.48550189  0.10126187  24.545 < 2.2e-16 ***
#> yeduc        0.11754673  0.00640225  18.360 < 2.2e-16 ***
#> exper        0.19617365  0.00831571  23.591 < 2.2e-16 ***
#> exper2      -0.00638115  0.00035017 -18.223 < 2.2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

coeftest の出力は、頑健な標準誤差（HC1）を用いて再計算した係数検定の結果です。係数の推定値自体は OLS と変わりませんが、標準誤差が変化しています（例: yeduc の標準誤差が 0.00706 から 0.00640 に減少）。この例では、標準誤差の変化は小さく、各変数の統計的有意性に関する結論は変わりませんでした。

なお、vcovHC() 関数を用いて分散共分散行列を計算する際に、type オプションを指定することで、異なる種類のロバスト標準誤差を計算することができます。例えば、type = "HC2" は、HC1 よりも一般的な推奨値として使用される種類のロバスト標準誤差を計算します。

頑健な F 検定（Wald 検定）

lmtest パッケージの waldtest 関数を用いて、頑健な標準誤差に基づいた F 検定（Wald 検定）を実行します。制約モデル reg0 と非制約モデル reg1 を、頑健な分散共分散行列 vcov を用いて比較します。

```r
# ミンサー方程式を OLS で重回帰
reg0 <- lm(lincome ~ yeduc, data = income6)
# 回帰の結果の確認
waldtest(reg0, reg1, vcov = vcov)
#> Wald test
#>
#> Model 1: lincome ~ yeduc
#> Model 2: lincome ~ yeduc + exper + exper2
#>   Res.Df Df      F    Pr(>F)
#> 1   4297
#> 2   4295  2 418.53 < 2.2e-16 ***
```

```
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

頑健な分散共分散行列を用いた Wald 検定の結果、F 統計量は 418.53 と非常に高く、p 値も極めて小さいです。これにより、`exper` と `exper2` が結合して統計的に有意であるという結論が、不均一分散を考慮した場合でも変わらないことが確認できます。

estimatr パッケージ

lm_robust による頑健な標準誤差の推定

`estimatr` パッケージの `lm_robust()` 関数は、回帰モデルの推定と同時に頑健な標準誤差の計算を行います。`se_type = "stata"`を指定することで、一段階で不均一分散を考慮した推定結果を得ることができます。

```
library(estimatr)
# ミンサー方程式を OLS で重回帰
reg_lm <- lm_robust(lincome ~ yeduc + exper + exper2, data = income6, se_type = "stata")
# 回帰の結果の確認
summary(reg_lm)
#>
#> Call:
#> lm_robust(formula = lincome ~ yeduc + exper + exper2, data = income6,
#>     se_type = "stata")
#>
#> Standard error type:  HC1
#>
#> Coefficients:
#>              Estimate Std. Error t value   Pr(>|t|)  CI Lower  CI Upper   DF
#> (Intercept)  2.485502  0.1012619   24.55 1.299e-124  2.286976  2.684027 4295
#> yeduc        0.117547  0.0064022   18.36  1.527e-72  0.104995  0.130098 4295
#> exper        0.196174  0.0083157   23.59 8.320e-116  0.179871  0.212477 4295
#> exper2      -0.006381  0.0003502  -18.22  1.574e-71 -0.007068 -0.005695 4295
#>
#> Multiple R-squared:  0.2066 ,    Adjusted R-squared:  0.206
#> F-statistic: 365.1 on 3 and 4295 DF,  p-value: < 2.2e-16
```

`lm_robust()` の出力結果は、標準誤差や t 値が `coeftest` と `vcovHC` を組み合わせた結果（ただし `se_type` による違いはある）と一致していることを示しています。これにより、`lm_robust()` が同様の頑健な推定をより直接的に行う便利な方法であることがわかります。

通常の標準誤差との比較

lm_robust() 関数で se_type = "classical"を指定すると、通常の OLS の標準誤差が計算されます。これを前のチャンクで計算した頑健な標準誤差の結果（reg_lm）と比較することで、不均一分散が標準誤差の推定に与える影響の大きさを具体的に確認できます。

```r
reg_0 <- lm_robust(lincome ~ yeduc + exper + exper2, data = income6, se_type = "classical")
# 回帰の結果の確認
summary(reg_0)
#>
#> Call:
#> lm_robust(formula = lincome ~ yeduc + exper + exper2, data = income6,
#>     se_type = "classical")
#>
#> Standard error type:  classical
#>
#> Coefficients:
#>              Estimate Std. Error t value   Pr(>|t|)  CI Lower  CI Upper   DF
#> (Intercept)  2.485502  0.1107823   22.44 1.636e-105  2.268311  2.702692 4295
#> yeduc        0.117547  0.0070603   16.65  2.311e-60  0.103705  0.131388 4295
#> exper        0.196174  0.0074935   26.18 2.754e-140  0.181482  0.210865 4295
#> exper2      -0.006381  0.0003162  -20.18  1.315e-86 -0.007001 -0.005761 4295
#>
#> Multiple R-squared:  0.2066 ,    Adjusted R-squared:  0.206
#> F-statistic: 372.8 on 3 and 4295 DF,  p-value: < 2.2e-16
```

このチャンクでは、se_type="classical"で計算したモデル reg_0 と、頑健標準誤差で計算した reg_lm を比較することを意図しています。summary(reg_0) を実行すれば、その標準誤差が lm() のデフォルト結果と一致し、summary(reg_lm) の頑健標準誤差との違いから不均一分散の影響を評価できます。

頑健なモデル間の F 検定

lm_robust() で推定したモデル同士を比較する場合も waldtest を使用します。se_type="stata"を指定して推定した制約モデル reg_lm0 と非制約モデル reg_lm を比較することで、頑健な標準誤差に基づいた F 検定を行います。

```r
reg_lm0 <- lm_robust(lincome ~ yeduc, data = income6, se_type = "stata")
# 回帰の結果の確認

waldtest(reg_lm0, reg_lm, test = "F")
#> Wald test
```

```
#>
#> Model 1: lincome ~ yeduc
#> Model 2: lincome ~ yeduc + exper + exper2
#>   Res.Df Df      F    Pr(>F)
#> 1   4297
#> 2   4295  2 418.53 < 2.2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

`lm_robust` で推定されたモデル同士を `waldtest` で比較した結果も、F 値は 418.53 と非常に有意です。これにより、頑健な標準誤差を用いたモデル間での比較検定も簡単に行え、結論が変わらないことが確認できます。

なお、ここでは `test = "F"` オプションを指定して F 検定を実行しています。`lm_robust` オブジェクトに対して `waldtest` を使用する場合、デフォルトではカイ二乗検定が行われます。対照的に、通常の `lm` オブジェクト同士を比較する場合のデフォルトは F 検定であり、カイ二乗検定を行うには `test = "Chisq"` を指定する必要があります。