

L'analyse de la variance.

Dans cet article nous allons reprendre les fondements théoriques de l'**ANalyse de la VAriance** (ANOVA) à un facteur et présenter une courte application : comparaison des temps moyens de production. L'analyse de la variance à **un facteur** est une méthode très répandue pour comparer au moins **3 sous-échantillons** d'après leur moyenne. Pour ce faire, l'anova va comparer la **variance inter-** et **intra-classe** afin de **savoir si au moins deux des moyennes diffèrent significativement**. L'anova repose donc sur un test de Fisher-Snedecor de comparaison de variance. Pourquoi utiliser un test de variance pour étudier les différences de moyenne ? Simplement parce que la dispersion des données peut avoir deux origines :

- la **variabilité liée au facteur** : la variance factorielle ou **inter-classe** ;
- la **variabilité intrinsèque à chaque catégorie du facteur** : la variance résiduelle ou **intra-classe** - la part de variabilité restante une fois la variabilité factorielle soustraite à la variabilité totale.

Ainsi, on va chercher à **savoir si la part de variance inter-classe est significativement supérieure à la part de variance intra-classe**. On pourra alors conclure qu'au moins deux des moyennes sont globalement différentes si la variabilité intra-class (résiduelle) est faible relativement à la part de variabilité inter-class (factorielle). Pour l'écrire différemment, plus le rapport des variances inter- et intra-classe sera élevé plus on aura de chance de conclure que **les moyennes de nos sous-échantillons sont significativement différentes de la moyenne générale**. L'étude de la variabilité de nos sous-échantillons va donc nous permettre d'étudier les divergences de moyennes.

La **statistique de test** sera donc le rapport entre la variance inter-classe et la variance intra-classe. Pour établir la significativité statistique de ce test on va utiliser le **test F de Fisher-Snedecor** qui suit une **loi de Fisher à $K-1$ et $N-K$ degrés de liberté** ; $K-1$ correspond au nombre de sous-échantillons moins un, $N-K$ correspond au nombre d'observations diminué du nombre de modalités du facteur explicatif.

$$F_{K-1; N-K} = \frac{\text{Variance}_{\text{factorielle}}}{\text{Variance}_{\text{résiduelle}}}$$

Après avoir fixé un **risque de première espèce** α , en général à 5%, si la statistique de test est supérieure à la valeur critique de la table de Fisher pour $K-1$ et $N-K$ degrés de liberté avec un seuil de significativité α , on conclura qu'**au moins une des moyennes diffère significativement d'une autre**.

Le **jeu des hypothèses** est le suivant :

H_0 : les moyennes des populations sont identiques ;

H_1 : Au moins deux populations parentes ne partagent pas la même moyenne.

Notez que ce test va être réalisé à partir d'un modèle linéaire. En effet, même si cela n'est pas évident, **l'anova est un modèle linéaire** de la forme suivante :

$$Y_{ij} = \beta_j + U_{ij}$$

où Y_{ij} est la réponse de l'observation i du groupe j , β_j le paramètre du groupe j à estimer et U_{ij} l'erreur associée à l'individu i du groupe j . L'erreur est identiquement et indépendamment distribuée, $U_{ij} \sim \mathcal{N}(0, \sigma^2)$ - i.e. **normalité et homoscedasticité sont requises en chaque point du facteur explicatif**.

Précisément, β_j correspond à la moyenne du groupe j et U_{ij} à l'écart entre l'observation i et la moyenne du groupe j auquel elle appartient. Avec ce modèle, pour un facteur à k modalités, le jeu des hypothèses peut s'exprimer :

- H_0 : $\beta_1 = \beta_2 = \dots = \beta_k$;
- H_1 : $\exists l \neq m : \beta_l \neq \beta_m$.

Sous l'hypothèse nulle, le **modèle** est dit **constant** (ou blanc) et se distingue du modèle considéré puisque le paramètre β n'est autre que la **moyenne globale** de la variable endogène. Alors, la statistique de test F va **comparer les différences de valeurs prédites entre le modèle considéré et le modèle blanc par rapport à l'écart-type** ; soit la somme des carrés factoriels (divisé par son degré de liberté, $K-1$) par rapport à la somme des carrés résiduels (divisé par son degré de liberté, $n-k$) ; soit le rapport entre la variance factorielle et la variance résiduelle^[1].

Take home messages.

- L'anova est un **modèle linéaire** employé pour **comparer les moyennes de plus de deux sous-échantillons** (les sous-échantillons formés par une variable quantitative relativement à un facteur qualitatif à plus de 2 modalités) ;
- Les hypothèses de **normalité**, d'**homoscédasticité** et d'**indépendance des résidus** doivent être satisfaites (de façon équivalente, les données doivent être normalement distribuées et à variance homogène en chaque point du facteur explicatif) ;
- Pour étudier les différences de moyennes, on utilise le rapport de la variance inter- et intra-classe : la significativité statistique d'au moins une différence est établie avec un test F de Fisher-Snedecor.

Dans cet article, pour chaque sortie de code, le code python et R est disponible.

Application : comparaison des temps moyens de production.

Aujourd'hui l'application va s'intéresser aux temps de production de trois chaînes d'assemblage. Plus précisément, on cherche à savoir si une des chaînes d'assemblage est plus rapide ou plus lente qu'une autre. On a donc relevé aléatoirement 100 mesures du temps de production pour les 3 chaînes.

Notre application repose sur des données fictives dont voici un court aperçu.

	chaîne_production	temps_production
0	A	36.014622
1	A	38.043625
2	A	35.204505
3	A	38.484877
4	A	37.948544

On dénombre **300 observations indépendantes** et deux variables, l'une quantitative continue (*temps_production*), l'autre qualitative à 3 modalités (*chaîne_production*). Voici une courte description de ces variables : la variable *temps_production* à une moyenne de 56.87 minutes et un écart-type de 18.94 minutes. Le minimum est de 25.18 minutes et le maximum de 98.99 minutes. Concernant la variable

chaîne_production, chaque modalité compte 100 observations. L'ensemble de données ne comporte aucune valeur manquante, ni valeurs extrêmes.

```
count    300.000000
mean     56.871202
std      18.938976
min      25.181439
25%      38.636396
50%      55.862853
75%      75.503349
max      98.989768
Name: temps_production, dtype: float64
```

Passons à la **réalisation de l'anova**. En **python** vous avez deux possibilités pour appliquer une *anova*, soit vous utiliser une fonction *f_oneway* du module python *scipy.stats*, soit vous utiliser un modèle linéaire généralisé sur lequel vous appliquer l'anova avec le module *statsmodels*. Dans le premier cas, vous ne pourrez pas accéder à la **table d'analyse de la variance**, vous obtiendrez la statistique de test et la pvalue. Voyons cela en pratique.

Dans un premier temps, nous allons ajuster le modèle linéaire afin de vérifier la satisfactions des hypothèses, donc la validité des résultats.

```
{% tabs ajustement %}
```

```
{% tab ajustement python %}
```

```
from statsmodels.formula.api import ols
# ajustement du modèle linéaire
model = ols(formula = 'temps_production ~ C(chaîne_production)',
data=df).fit()

# si vous souhaitez afficher les résultats de la regression linéaire
# print(model.summary())

# on récupère les erreurs du modèle
residuals = model.resid
```

Le module *statsmodels.formula.api* permet de déclarer un modèle OLS à la R.

```
{% endtab %}
```

```
{% tab ajustement r %}
```

```
library(stats)

# ajustement du modèle linéaire
model <- lm(temps_production ~ factor(chaîne_production), data = df)
```

```
# afficher les résultats de la régression linéaire
# summary(model)

# on récupère les erreurs du modèle
residuals <- residuals(model)
```

```
{% endtab %} {% endtabs %}
```

Vérification des conditions d'application de l'anova.

Normalité des résidus.

```
{% tabs normalite_residu %}
```

```
{% tab normalite_residu python %}
```

```
from scipy import stats

# on vérifie l'hypothèse de normalité avec un test d'adéquation
paramétrique
shapiro_test = stats.shapiro(residuals)

print("Test de Shapiro-Wilk :")
print(f"Statistique : {shapiro_test.statistic}, p-value :
{shapiro_test.pvalue}")

    Test de Shapiro-Wilk :
    Statistique : 0.9907482862472534, p-value : 0.05559874325990677

# on complète ce test d'une observation graphique
import statsmodels.api as sm

sm.qqplot(residuals, line='s')
plt.title('Q-Q Plot des résidus')
plt.show()
```

```
{% endtab %}
```

```
{% tab normalite_residu r %}
```

```
library(stats)

# on vérifie l'hypothèse de normalité avec un test d'adéquation
paramétrique
shapiro_test <- shapiro.test(residuals)
cat("Test de Shapiro-Wilk :\n")
cat("Statistique :", shapiro_test$statistic, ", p-value :",
shapiro_test$p.value, "\n")
```

```
Test de Shapiro-Wilk :
Statistique : 0.9907482862472534, p-value : 0.05559874325990677
```

```
# on complète ce test d'une observation graphique
qqnorm(residuals, main = "Q-Q Plot des résidus", xlab = "Quantiles
théoriques", ylab = "Quantiles des résidus")
qqline(residuals, col = "red")
```

```
# ou simplement,
# plot(model, 2)
```

```
{% endtab %}
```

```
{% tab normalite_residu QQ-plot %}
```

```

```

```
{% endtab %}
```

```
{% endtabs %}
```

D'après le test de Shapiro-Wilk nous n'avons pas suffisamment d'évidence pour rejeter l'hypothèse nulle de normalité des résidus. Le graphique quantiles-quantiles met en évidence la présence de quelques outliers ; globalement, l'adéquation à la loi Normale est bonne.

Homoscédasticité des résidus.

Pour vérifier l'homogénéité des variances on va utiliser le test de Breusch-Pagan ainsi qu'un graphique des résidus en fonction des valeurs prédites.

```
{% tabs homoscedasticite_residu %}
```

```
{% tab homoscedasticite_residu python %}
```

```
import statsmodels.stats.api as sms

# test de Breusch-Pagan pour l'homoscédasticité
bp_test = sms.het_breuschpagan(residuals, model.model.exog)
labels = ['LM Stat', 'LM p-value', 'F Stat', 'F p-value']
bp_results = dict(zip(labels, bp_test))
print("Test de Breusch-Pagan :")
for key, value in bp_results.items():
    print(f"{key} : {value}")

Test de Breusch-Pagan :
LM Stat : 0.9712142537018131
LM p-value : 0.6153235039150617
F Stat : 0.4823124847822556
F p-value : 0.6178368047840446
```

```
# graphique des résidus vs valeurs ajustées
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
plt.scatter(model.fittedvalues, residuals, alpha=0.5)
plt.axhline(y=0, color='r', linestyle='--')
plt.title('Résidus par rapport aux valeurs prédites')
plt.xlabel('Valeurs prédites')
plt.ylabel('Résidus')
plt.show()
```

```
{% endtab %}
```

```
{% tab homoscedasticite_residu r %}
```

```
library(lmtest)
library(ggplot2)

# test de Breusch-Pagan pour l'homoscédasticité
bp_test <- bptest(df$$temps_production ~ df$$chaîne_production, data = df)
print(bp_test)

studentized Breusch-Pagan test

data: df$$temps_production ~ df$$chaîne_production
BP = 5.8215, df = 2, p-value = 0.05443

# graphique des résidus vs valeurs ajustées
residuals <- resid(model)
fitted_values <- fitted(model)

residuals_df <- data.frame(Fitted = fitted_values, Residuals = residuals)

ggplot(residuals_df, aes(x = Fitted, y = Residuals)) +
  geom_point(alpha = 0.5, color = "blue") +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  labs(title = "Résidus par rapport aux valeurs prédites",
       x = "Valeurs prédites",
       y = "Résidus") +
  theme_minimal()

# ou simplement,
# plot(model, 1)
```

```
{% endtab %}
```

```
{% tab homoscedasticite_residu residual(fitted)-plot %}
```

```

```

```
{% endtab %}
```

```
{% endtabs %}
```

Le test de Breusch-Pagan confirme l'homogénéité des variances : on ne peut pas rejeter l'hypothèse nulle. Également, le graphique montre une distribution homogène des observations en chaque point du facteur explicatif.

Indépendance des données.

Il nous reste à vérifier l'indépendance des données, on peut utiliser le test de Durbin-Watson. Notez que l'indépendance des données est généralement garantie par la façon dont sont recueillies les données.

```
{% tabs independance_residu %}
```

```
{% tab independance_residu python %}
```

```
from statsmodels.stats.stattools import durbin_watson

# test de Durbin-Watson
dw_statistic = durbin_watson(residuals)
print(f"Statistique de Durbin-Watson : {dw_statistic}")
```

Statistique de Durbin-Watson : 2.1853337338421053

```
{% endtab %}
```

```
{% tab independance_residu r %}
```

```
library(lmtest)

# test de Durbin-Watson
durbin_watson <- dwtest(model)
print(durbin_watson)
```

Durbin-Watson test

data: model
DW = 2.1853, p-value = 0.6042
alternative hypothesis: true autocorrelation is greater than 0

```
{% endtab %}
```

```
{% endtabs %}
```

La statistique de Durbin-Watson varie entre 0 et 4. Une valeur de 2 garantit l'indépendance des données. Qui plus est, le test avec R donne la pvalue : les erreurs du modèle sont indépendantes.

Nous avons donc vérifier l'ensemble des conditions d'application de l'anova. Avant de passer à la lecture des résultats, voyons une autre façon de vérifier les hypothèses de validité de l'Anova.

Validité des hypothèses.

Je disais précédemment que "normalité et homoscedasticité sont requises en chaque point du facteur explicatif." En effet, cela est équivalent à dire que les résidus sont normalement distribuée et à variance homogène. Pour vous en convaincre, observez les diagrammes en boîte suivant : un diagramme pour chaque sous-échantillon.

```
{% tabs graphique_distribution %}
```

```
{% tab graphique_distribution python %}
```

```
import plotly.express as px

fig = px.box(data, x = "chaîne_production", y="temps_production")
fig.show()
```

```
{% endtab %}
```

```
{% tab graphique_distribution r %}
```

```
library(ggplot2)

ggplot(df, aes(x = chaîne_production, y = temps_production)) +
  geom_boxplot() +
  labs(title = "Diagramme en boîte de temps de production par chaîne",
       x = "Chaîne de production",
       y = "Temps de production") +
  theme_minimal()
```

```
{% endtab %}
```

```
{% endtabs %}
```

![Diagrammes en boîte - distribution de la variable dépendante en chaque point du facteur explicatif]({{ site.baseurl }}/assets/img/anova_temps_production/boxplot.png "Distribution de Y relativement à X.")

Sur ce graphique on peut observer la **distribution de la variable dépendante en chaque point du facteur explicatif**. Premièrement, on note que les distributions sont à variance homogène. Deuxièmement, les distributions sont normales malgré la présence d'un outlier parmi les chaînes de production B et C. Nous allons maintenant tester la significativité statistique de nos hypothèses.

```
{% tabs normalite_distribution %}
```

```
{% tab normalite_distribution python %}
```



```

from scipy import stats
# tests d'hypothèses en chaque modalité du facteur explicatif

# groupement des valeurs par chaîne de production
groups = df.groupby("chaîne_production")["temps_production"].apply(list)

# normalité en chaque point du facteur explicatif
print("Normalité de la variable en chaque point du facteur explicatif")
print("---"*35)
for group in groups:
    statistic, pvalue = stats.shapiro(group)
    print(f"group has statistic {statistic} and pvalue {pvalue}")

print("---"*35)

# homoscedasticité en chaque point du facteur explicatif

statistic, pvalue = stats.bartlett(*groups)
print("Homogénéité des variances avec le test de Bartlett : ")
print("---"*35)
print(f"The statistic is {statistic} and the pvalue {pvalue}")

```

Voici les résultats des tests statistiques.

```

Normalité de la variable en chaque point du facteur explicatif
-----
-----
group has statistic 0.9883927202932101 and pvalue 0.5375996303014767
group has statistic 0.9817484541115354 and pvalue 0.18149418636596398
group has statistic 0.9889764351567896 and pvalue 0.5827173779582634
-----
-----
Homogénéité des variances avec le test de Bartlett :
-----
-----
The statistic is 1.9816262409012764 and the pvalue 0.37127467721903507

```

```
{% endtab %}
```

```
{% tab normalite_distribution r %}
```

```

library(lmtest)

# normalité en chaque point du facteur explicatif
cat("Normalité de la variable en chaque point du facteur explicatif\n")
cat(strrep("-", 35), "\n")

for (level in unique(df$$chaîne_production)) {
  group_data <- df %>% filter(chaîne_production == level) %>%

```

```
pull(temps_production)
shapiro_test <- shapiro.test(group_data)
cat(sprintf("Groupe '%s' a une statistique %f et p-value %f\n", level,
shapiro_test$statistic, shapiro_test$p.value))
}

cat(strrep("-", 35), "\n")

# homoscedasticité en chaque point du facteur explicatif
library(stats)

bartlett_test <- bartlett.test(temps_production ~ chaine_production, data =
df)
cat("Homogénéité des variances avec le test de Bartlett : \n")
cat(strrep("-", 35), "\n")
cat(sprintf("La statistique est %f et la p-value %f\n",
bartlett_test$statistic, bartlett_test$p.value))
```

Voici les résultats du test.

```
Normalité de la variable en chaque point du facteur explicatif
-----
Groupe 'A' a une statistique 0.993878 et p-value 0.934929
Groupe 'B' a une statistique 0.972886 et p-value 0.181494
Groupe 'C' a une statistique 0.976043 et p-value 0.065127
-----
Homogénéité des variances avec le test de Bartlett :
-----
La statistique est 5.380940 et la p-value 0.067849
```

```
{% endtab %}
```

```
{% endtabs %}
```

Les données de la variable dépendante sont donc normalement distribuées en chaque point du facteur explicatif et à variances homogènes. Les conclusions sont équivalentes à celles des tests effectués sur les résidus du modèle linéaire qu'on a ajusté en début d'application. Pour cause, les deux assertions "les résidus sont normalement distribués et à variance homogène" et "les données sont normalement distribués et à variances homogènes en chaque point du facteur explicatif" sont équivalentes.

J'utilise toujours un boxplot conditionnel avant de réaliser les tests d'adéquation et d'homogénéité des résidus. Cela permet d'avoir un premier visuel de la satisfaction des hypothèses d'application.

Anova - Calcul de la statistique de test F.

Voyons maintenant comment calculer notre statistique de test F. On sait qu'elle est le rapport de la variance factorielle et résiduelle. On doit donc calculer ces variances. Pour cela, on va utiliser les paramètres du modèle linéaire ajusté en début d'application.

```
{% tabs variances %}
```

```
{% tab variances python %}
```

```
# somme des carrés résiduels (SSE)
SSE = sum((model.resid) ** 2)

# somme des carrés totaux (SST) = Variabilité totale
SST = sum((df['temps_production'] - df['temps_production'].mean()) ** 2)

# somme des carrés factoriels (SSF)
SSF = SST - SSE

# degrés de liberté
dfn = model.df_model
dfd = model.df_resid

print(f"La somme des carrés factoriels {SSF} divisée par le degré de
liberté factoriels {dfn} donne la variance factorielle {SSF/dfn}.")
print(f"La somme des carrés résiduels {SSE} divisée par le degré de liberté
résiduel {dfd} donne la variance résiduelle {SSE/dfd}.")
```

La somme des carrés factoriels 103876.91142829404 divisée par le degré de liberté factoriels 2.0 donne la variance factorielle 51938.45571414702.

La somme des carrés résiduels 9069.353797436608 divisée par le degré de liberté résiduel 297.0 donne la variance résiduelle 30.53654477251383.

```
{% endtab %}
```

```
{% tab variances r %}
```

```
# somme des carrés résiduels (SSE)
SSE <- sum(model$residuals^2)

# somme des carrés totaux (SST) = Variabilité totale
SST <- sum((df$temps_production - mean(df$temps_production))^2)

# somme des carrés factoriels (SSF)
SSF <- SST - SSE

# degrés de liberté
dfn <- length(coef(model)) - 1
dfd <- df.residual(model)

cat(sprintf("La somme des carrés factoriels %f divisée par le degré de
liberté factoriels %f donne la variance factorielle %f.\n",
            SSF, dfn, SSF / dfn))
cat(sprintf("La somme des carrés résiduels %f divisée par le degré de
liberté résiduel %f donne la variance résiduelle %f.\n",
            SSE, dfd, SSE / dfd))
```

La somme des carrés factoriels 103876.91142829404 divisée par le degré de liberté factoriels 2.0 donne la variance factorielle 51938.45571414702.
 La somme des carrés résiduels 9069.353797436608 divisée par le degré de liberté résiduel 297.0 donne la variance résiduelle 30.53654477251383.

```
{% endtab %}
```

```
{% endtabs %}
```

On peut maintenant calculer notre statistique de test F, c'est simplement le rapport de la variance factorielle et résiduelle.

```
{% tabs statistiqueF %}
```

```
{% tab statistiqueF python %}
```

```
# calcul de la statistique F
variance_factorielle = SSF/dfn
variance_residuelle = SSE/dfd
statistique_F = variance_factorielle/variance_residuelle

print(f"La statistique de test est : F={statistique_F}")

La statistique de test est : F=1700.862232484705
```

On peut vérifier notre résultat grâce à la statistique F du modèle linéaire ajusté précédemment :

```
statistique_F = model.fvalue
print(f"La statistique F est : {statistique_F}")

La statistique F est : 1700.8622324847042
```

```
{% endtab %}
```

```
{% tab statistiqueF r %}
```

```
variance_factorielle <- SSF / dfn
variance_residuelle <- SSE / dfd

statistique_F <- variance_factorielle / variance_residuelle
cat(sprintf("La statistique de test est : F=%f", statistique_F))

La statistique de test est : F=1700.862232

# vérification du calcul à partir des données du model ajusté
statistique_F <- summary(model)$fstatistic[1]
cat(sprintf("La statistique F est : %f", statistique_F))
```

La statistique F est : 1700.862232

{% endtab %}

{% endtabs %}

Il ne nous reste plus qu'à comparer cette statistique au quotient de la table de Fisher pour 2 et 297 degrés de liberté. Avec python, on peut retrouver cette valeur critique grâce à la fonction `stats.f.ppf` du module `scipy`. Cette fonction prend en paramètre le **niveau de confiance** ($1-\alpha$), le nombre de **degrés de liberté** du numérateur (dfn) et du dénominateur (dfd). Avec R, vous pouvez utiliser la fonction `qf` du package standard `stats`; les paramètres sont identiques.

{% tabs quotient %}

{% tab quotient python %}

```
from scipy import stats

# recherche quantile de la loi de Fisher
alpha = 0.05
quantile_critique = stats.f.ppf(1 - alpha, dfn, dfd)

print(f"Statistique F du modèle : {statistique_F}")
print(f"Quantile critique à {1-alpha} pour Fisher({int(dfn)}, {int(dfd)}) : {quantile_critique}")

Statistique F du modèle : 1700.8622324847042
Quantile critique à 0.95 pour Fisher(2, 297) : 3.0261533685653728

# règle de décision
if statistique_F > quantile_critique:
    print("La statistique F est significative, on rejette l'hypothèse nulle.")
else:
    print("La statistique F n'est pas significative, on ne rejette pas l'hypothèse nulle.")

La statistique F est significative, on rejette l'hypothèse nulle.
```

{% endtab %}

{% tab quotient r %}

```
library(stats)

alpha <- 0.05
```

```

quantile_critique <- qf(1 - alpha, dfn, dfd)

cat(sprintf("Statistique F du modèle : %f\n", statistique_F))
cat(sprintf("Quantile critique à %f pour Fisher(%d, %d) : %f\n", 1 - alpha,
as.integer(dfn), as.integer(dfd), quantile_critique))

Statistique F du modèle : 1700.8622324847042
Quantile critique à 0.95 pour Fisher(2, 297) : 3.0261533685653728

if (statistique_F > quantile_critique) {
  cat("La statistique F est significative, on rejette l'hypothèse
nulle.")
} else {
  cat("La statistique F n'est pas significative, on ne rejette pas
l'hypothèse nulle.")
}

La statistique F est significative, on rejette l'hypothèse nulle.

```

```
{% endtab %}
```

```
{% endtabs %}
```

Si vous souhaitez trouver la probabilité d'observer une statistique de test au moins aussi grande - c'est-à-dire la **pvalue** - vous devez utiliser la **fonction cumulative** de la loi de Fisher (fonction de répartition cumulative). Précisément, on cherche la probabilité d'observer une statistique de test au moins aussi grande que celle du test. Il faut donc trouver le complément de la probabilité retournée par la fonction de répartition cumulative, soit $\alpha = P(X \geq \text{statistique_F}) = 1 - P(X \leq \text{statistique_F})$.

```
{% tabs pvalue %}
```

```
{% tab pvalue python %}
```

```

from scipy import stats

# calcul de la pvalue
pvalue = 1 - stats.f.cdf(statistique_F, dfn, dfd)
print(f"La pvalue de la statistique de test F est : {pvalue}")

La pvalue de la statistique de test F est : 2.220446049250313e-16

```

```
{% endtab %}
```

```
{% tab pvalue r %}
```

```

library(stats)

pvalue <- 1 - pf(statistique_F, dfn, dfd)
cat(sprintf("La p-value de la statistique de test F est : %f\n", pvalue))

```

La pvalue de la statistic de test **F** est : **2.220446049250313e-16**

{% endtab %}

{% endtabs %}

Puisque notre pvalue est inférieure à 5%, on conclut au rejet de l'hypothèse nulle : **nous avons suffisamment d'évidence pour affirmer qu'au moins deux populations parentes ont des moyennes significativement différentes.**

Pour plus d'informations sur le [test F de Fisher](#) ou le calcul du quotient et de la pvalue, [voici une référence](#).

Anova - table d'analyse de la variance.

Voici le table d'analyse de la variance. Elle présente d'une part la somme des carrés (**sum_sq**) factoriel et résiduels, d'autre part les nombres de degrés de liberté (**df**). Puis, on retrouve la statistique de test **F** et la pvalue associée à cette statistique (**PR(>F)**).

{% tabs table_variance %}

{% tab table_variance python %}

```
# application de l'ANOVA
anova_table = sm.stats.anova_lm(model, typ=2)
print(anova_table)
```

	sum_sq	df	F	PR(>F)
C(chaine_production)	103876.911428	2.0	1700.862232	2.231586e-163
Residual	9069.353797	297.0	NaN	NaN

{% endtab %}

{% tab table_variance r %}

```
library(car)
anova_table <- Anova(model, type=2)
print(anova_table)
```

Anova Table (Type II tests)

Response: temps_production

	Sum Sq	Df	F value	Pr(>F)
factor(chaine_production)	103876.911428	2	1700.862	2.231586e-163
Residual	9069.353797	297		

Signif. codes: 0 '****' 0.001 '***' 0.01 '**' 0.05 '.' 0.1 ' ' 1

{% endtab %}

{% endtabs %}

Les résultats de la table d'analyse de la variance sont similaires à ceux que nous avons calculés à la section précédente. Pour retrouver la statistique de test, nous l'avons vu, il vous suffit de diviser la somme des carrés avec leur degré de liberté respectif puis de faire le rapport de la variance factorielle et de la variance résiduelle.

$$Variance_factorielle = \frac{\text{sum_sq}(\text{chaîne_production})}{df(\text{chaîne_production})} = \frac{103876.911428}{2.00} = 51938.4557$$

$$Variance_résiduelle = \frac{\text{sum_sq}(\text{Residual})}{df(\text{Residual})} = \frac{9069.353797436608}{297.00} = 30.5365$$

$$statistique_F = \frac{variance_factorielle}{variance_résiduelle} = 1700.862232$$

D'après la pvalue de la statistique de test, **il est peu probable que toutes les populations parentes aient une moyenne identique**. Autrement dit, il est probable qu'au moins une machine présente un temps d'assemblage significativement différent d'une autre. Pour savoir laquelle, il faut réaliser un test post-hoc afin de comparer tous les sous-échantillons deux-à-deux ou à un groupe de référence.

Voilà, nous sommes arrivés au bout de cette note sur l'analyse de la variance. Dans un prochain billet, nous verrons l'application des tests post-hoc tels que la procédure de Dunnett et de Tukey. À bientôt !

Si vous souhaitez une application des tests post-hoc, vous pouvez consulter mon article sur l' [analyse des prix moyens au mètre carré des logements de la ville de Lyon](#) .

[^1]: Pour plus de détail sur le modèle linéaire Anova, [consultez cet article](#).