Module 4. Data analysis techniques

In this module, we will review different techniques in data analysis and how to pick the proper technique based on the question asked, the data type, and the data size available. At the end, we will do a case study of data analysis, starting with a question.

Unit 4.1 Statistical analysis techniques

Statistical analysis forms the backbone of data analysis, providing rigorous methods for understanding relationships, testing hypotheses, and making predictions from data. In the previous modules, we discussed descriptive statistics like mean, median, variance, standard deviation, quartiles, coefficient of variation, and hypothesis testing in detail. We now examine ANOVA, regression, and correlation in depth.

1. **Analysis of variance (ANOVA)**

ANOVA is a statistical method used to analyze differences among means of multiple groups. It extends the t-test concept to situations with more than two groups by comparing the variance between the groups to the variance within the groups.

The fundamental principle relies on separating the total variance in the data into:

1. between-group variance (explained variance),
2. within-group variance (error/unexplained variance).

The key mathematical components are:

SST (total sum of squares) = SSB (between sum of squares) + SSW (within sum of squares).

Where:

- SST measures total variability in the data.
- SSB measures variability between group means.
- SSW measures variability within groups.

Types of ANOVA

1. **One-way ANOVA**

   One-way ANOVA is the simplest form of ANOVA. It is a statistical test used to compare means between three or more independent groups. It extends the principles of a t-test, which can only compare two groups. It examines the effect of one independent variable on the dependent variable

   Hypothesis

- Null hypothesis(H0): all groups are equal ($\mu_1 = \mu_2 = \mu_3$...).
- Alternative (H1): at least one group's mean is different from the others.

Variance

- Total variance (SST) = overall variability in the entire dataset.
- Between-group variance (SSB) = variability between group means.
- Within-group Variance (SSW) = variability within each group.

F-Statistic - measures whether a set of independent variables jointly significantly affects the dependent variable.

F = (SSB/df1) / (SSW/df2)

df1 = number of predictors (p)
df2 = sample size (n) - number of predictors (p) - 1

- Larger F-values suggest more significant differences between groups.

Example: let us use an example from tennis to explain how to apply one-way ANOVA.

Question: Is there a significant difference in serve speeds (mph) between professional tennis players from different tour levels (ATP, Challenger, and Futures)?

Data: serve speeds data from ATP, Challenger and Futures tours.

ATP Tour (Group 1): 125, 128, 130, 122, 127, 129, 124, 126
Challenger Tour (Group 2): 120, 122, 119, 124, 121, 118, 123, 120
Futures Tour (Group 3): 115, 118, 117, 120, 116, 119, 114, 117

Let us perform ANOVA step-by-step.

1. Calculate the means of each group

ATP mean: 126.375 mph
Challenger mean: 120.875 mph
Futures mean: 117.0 mph.
Grand mean = 121.417 mph

2. Calculate the sum of squares

Between groups (SSB):

Sum of (ni × (group mean - grand mean)^2)
SSB = 8(126.375 - 121.417)^2 + 8(120.875 - 121.417)^2 + 8(117.000 - 121.417)^2
SSB = 432.08

Within groups (SSW):

The sum of (individual value - group mean)^2 for each group
SSW = 56.875 + 48.875 + 46.000
SSW = 151.75

3. Calculate degrees of freedom

   df between = number of groups – 1 = 2
   df within = total observations – number of groups = 21

4. Calculate F-Statistic
   F = (SSB/df between) / (SSW/df within) F = (432.08/2) / (151.75/21) F = 29.83

5. Decision At $\alpha = 0.05$, critical $F(2,21) \approx 3.47$ Since $29.83 > 3.47$, reject null hypothesis

   This means that significance level (α) = 0.05 (5% chance of Type 1 error).
   Degrees of freedom: 2 & 21 (F(2,21)
   Critical F-value = 3.47
   Calculated F-value = 29.83.

   If the calculated F-value > critical F-value, then Reject the null hypothesis.

6. Conclusion: there is strong statistical evidence ($F(2,21) = 29.83$, $p < .001$) that serve speeds differ significantly between tour levels. Post-hoc analysis reveals that ATP players serve fastest, followed by Challenger players, with Futures players serving slowest, with all differences being statistically significant. The chance that these differences are random is less than 5 %.

7. Practical significance
   This analysis helps:
   - coaches set appropriate training targets,
   - players understand performance benchmarks,
   - tournament organizers plan equipment and facilities,
   - scouts evaluate player development potential.

2. Two-way ANOVA

Two-way ANOVA examines how two independent variables (factors) affect a dependent variable, including their potential interaction effects. Unlike one-way ANOVA, it can detect both main effects and interaction effects.

There are three sets of hypotheses.

- Factor A main effect (H0: no effect of Factor A).
- Factor B main effect (H0: no effect of Factor B).
- Interaction effect (H0: no interaction between A & B).

The main effects are the independent impacts of each factor. Interaction effects are how factors influence each other.
Example: how do training methods (strength vs. endurance) and playing positions (forward vs. defender) affect sprint speed in soccer players?

Experiment design:
Factor A: training method (strength, endurance).
Factor B: playing position (forward, defender).
Dependent variable: 40-yard sprint time (seconds).

Data – sprint times:

- Forwards: 4.8, 4,9, 4.7, 4.8
- Defenders: 5.2, 5.3, 5.4, 5.3

Analysis steps:

1. Calculate cell means.
2. Calculate main effects.
   Training method:
   - Strength mean: 4.95
   - Endurance mean: 5.15
   - Difference: 0.2s

   Position:

   - Forward mean: 4.9
   - Defender mean 5.2
   - Difference: 0.3
3. Calculate F-Statistics.

   Training method: $F(1,12) = 25.6$, $p < 0.001$
   Position: $F(1,12) = 56.3$, $p < 0.001$
   Interaction: $F(1,12) = 0.0$, $p = 1.0$

4. Results and interpretation.
   - Training method: significant effect ($p < 0.001$)
     - Strength training leads to faster sprint times.
   - Position: significant effect($p < 0.001$)
     - Forwards are generally faster than defenders.
   - Interaction effect: no significant effect ($p = 1.0$)
     - The effect of the training method is consistent across positions.

5. Applications. This analysis shows how two-way ANOVA can help coaches and teams:
    - optimize training programs,
    - make evidence-based decisions,
    - understand player development better.

3. **Repeated measures ANOVA**
   Repeated measures ANOVA (RM-ANOVA) is used when the same subjects are measured multiple times under different conditions or time points. It accounts for the correlation between repeated measurements within subjects.

   RM-ANOVA is usually applied to answer the following types of questions:

    - How does fatigue affect basketball free throw accuracy across four quarters in NBA players?
    - What is the impact of altitude training on VO2 max levels tested pre-camp, mid-camp, post-camp, and 2 weeks after return?
    - How does pitching velocity change across nine innings for baseball pitchers?

Unit 4.2 Introduction to regression analysis

Regression analysis is a powerful statistical method for modeling relationships between variables. It enables prediction, inference, and understanding of complex relationships in data.

4.2.1 Linear regression

1. **Simple linear regression**
    - Model form: $Y = \beta_0 + \beta_1 X + \varepsilon$
    - **Components:**
        - Dependent variable (Y)
        - Independent variable (X)
        - Intercept (β_0)
        - Slope coefficient (β_1)
        - Error term (ε)
    - **Assumptions**
        - Linearity: relationship is linear in parameters.
        - Independence: observations are independent.
        - Normality: errors are normally distributed.
    - **Key metrics**
        - R^2: proportion of variance explained (0 to 1).

- p-value: statistical significance.
- Standard error: uncertainty in coefficient estimates.
- Residuals: differences between predicted and actual values.

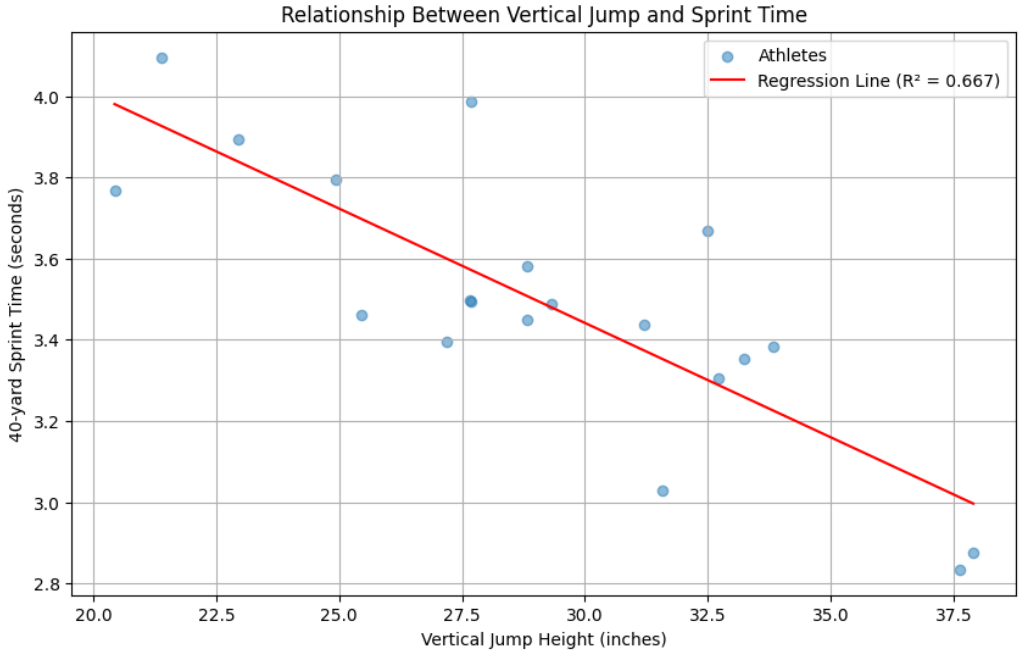
Example: relationship between jump height and sprint speed in athletes.

Data: we created random jump height data (in inches) and 40-yard sprint time (seconds) for 20 athletes.

The Python code below

- Draws a scatterplot of all the raw data points.
- The red line shows the linear relationship between sprint time and jump height.
- R2 (R-squared) tells us how well jump height predicts sprint time.

Figure 1. Relationship between jump height and sprint speed in athletes



```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy import stats

# Create sample data for 20 athletes
np.random.seed(42)

# Vertical jump heights (inches)
jump_height = np.random.normal(30, 5, 20)

# Sprint speeds (seconds for 40-yard dash)
# Adding negative correlation (higher jumps = faster sprints)
sprint_time = 5.0 - 0.05 * jump_height + np.random.normal(0, 0.2, 20)

# Create DataFrame
data = pd.DataFrame({
    'Jump_Height': jump_height,
    'Sprint_Time': sprint_time
})

# Perform linear regression
slope, intercept, r_value, p_value, std_err = stats.linregress(data['Jump_Height'],
                                                               data['Sprint_Time'])

# Create scatter plot
plt.figure(figsize=(10, 6))
plt.scatter(data['Jump_Height'], data['Sprint_Time'], alpha=0.5, label='Athletes')

# Add regression line
x_line = np.array([data['Jump_Height'].min(), data['Jump_Height'].max()])
y_line = slope * x_line + intercept
plt.plot(x_line, y_line, color='red', label=f'Regression Line (R² = {r_value**2:.3f})')

# Customize plot
plt.title('Relationship Between Vertical Jump and Sprint Time')
plt.xlabel('Vertical Jump Height (inches)')
plt.ylabel('40-yard Sprint Time (seconds)')
plt.grid(True)
plt.legend()
```

```python
# Print results
print("Linear Regression Results:")
print(f"Slope: {slope:.4f}")
print(f"Intercept: {intercept:.4f}")
print(f"R-squared: {r_value**2:.3f}")
print(f"P-value: {p_value:.4f}")

# Example predictions
new_jumps = np.array([25, 30, 35])
predictions = slope * new_jumps + intercept

print("\nPredicted Sprint Times:")
for jump, pred in zip(new_jumps, predictions):
    print(f"Jump Height: {jump} inches -> Predicted Sprint Time: {pred:.2f} seconds")
```

Source: own elaboration.

Linear regression results

Slope: -0.0563

Intercept: 5.1317

R-squared: 0.667
P-value: 0.0000

Predicted sprint times
Jump Height: 25 inches -> Predicted Sprint Time: 3.72 seconds
Jump Height: 30 inches -> Predicted Sprint Time: 3.44 seconds
Jump Height: 35 inches -> Predicted Sprint Time: 3.16 seconds

Interpreting the results

- The negative slope indicates higher jumps = faster sprint times.
- R-squared of 0.667 shows a strong relation between the two.

This analysis will help coaches

- Predict sprint times based on jump tests.
- Identify athletes who are faster/slower than predicted.
- Use jumping ability as one of the strong indicators of sprint potential.
- Prescribe athletes the appropriate program to improve their sprint times.

- **Multiple linear regression**

Multiple linear regression extends simple linear regression by including various independent variables (predictors) to predict a dependent variable. This process works very similarly to the one described above for simple linear regression.

Examples of applications of multiple linear regression in sports

Predicting team wins in the NBA using:

- points scored per game,
- points allowed,
- turnover ratio,
- strength of schedule.

Soccer/football match outcome prediction using:

- possession percentage,
- shots on target,
- pass completion rate,
- distance covered,
- home/away status.

4.2.2 Advanced regression techniques

Various advanced regression techniques exist, such as logistic, poisson and ridge. We won't discuss all of these in detail, but we will describe one of the most popular techniques: logistic regression.

Logistic regression

Logistic regression is used for binary classification problems (outcomes with two possible values). Unlike linear regression, it predicts the probability of an event occurring.

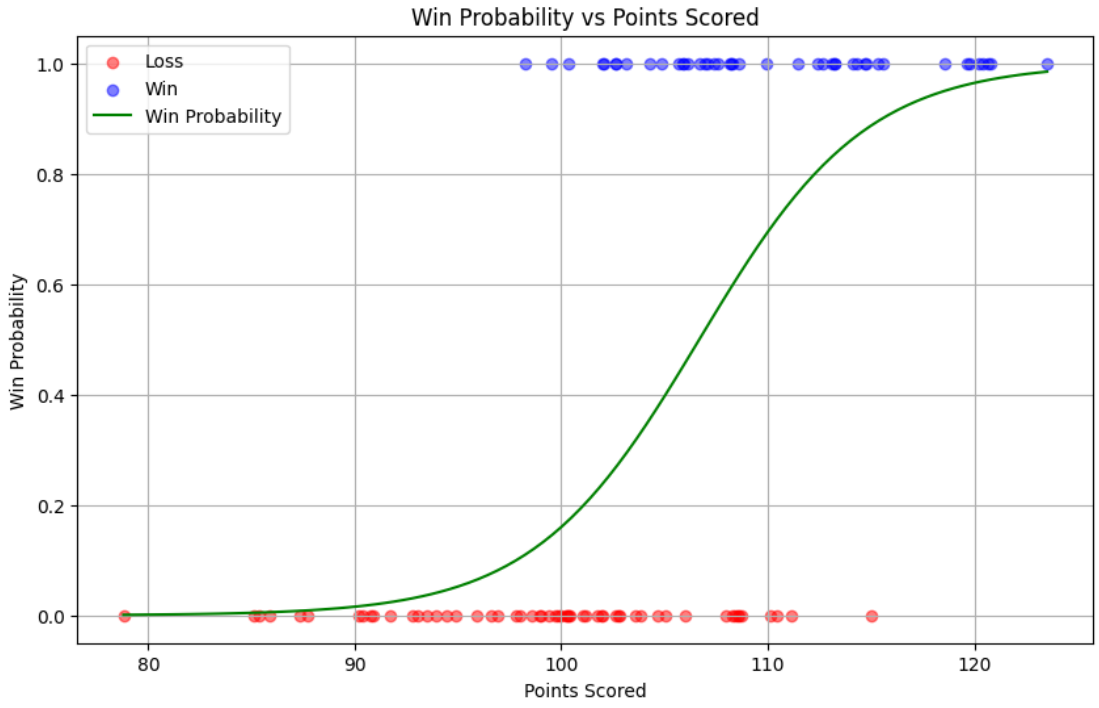
Key features

- Transforms linear input to probability (0 to 1).
- Output is always between 0 and 1.
- Output is an S-shaped curve.
- It is primarily used for classification problems.
    - Is the color of the ball red or white?
    - Is that a goal or not?
- Metrics to evaluate a logistic regression model.
    - Accuracy = correct predictions/total predictions.
    - Precision = true positives/(true positives + false positives).
    - Recall = true positives (true positives + false negatives).
    - ROC curve = True positive rate vs False positive rate
        - An ROC (receiver operating characteristic) curve plots the true positive rate (sensitivity) against the false positive rate (1-specificity) at various classification thresholds. It visualizes the performance of a binary classifier across all possible decision thresholds.

Example: predicting a basketball game using points scored as the predictor. We use sample data from 100 games.

Figure 2. Predicting a basketball game using points scored as the predictor

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import seaborn as sns

# Generate sample data for 100 games
np.random.seed(42)
n_games = 100

# Feature: Points scored
points_scored = np.random.normal(105, 10, n_games)

# Create win/loss based on points (with some randomness)
# Teams scoring more typically win more
probability = 1 / (1 + np.exp(-(points_scored - 105) / 5))
wins = np.random.binomial(n=1, p=probability)

# Create DataFrame
data = pd.DataFrame({
    'Points': points_scored,
    'Win': wins
})

# Split data into training and testing sets
X = data[['Points']]
y = data['Win']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Fit logistic regression model
model = LogisticRegression()
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)
y_pred_prob = model.predict_proba(X_test)[:, 1]
```



Source: own elaboration.

Visualization

The scatter plot shows

- Actual outcomes (red and blue dots).
- The probability S-curve showing the relationship between points scored and the outcome of the games.
- A clear decision boundary at 0.5 probability.

Code for printing results

Figure 3. Code for printing results

```
# Print model results
print("Model Coefficients:")
print(f"Intercept: {model.intercept_[0]:.3f}")
print(f"Points Coefficient: {model.coef_[0][0]:.3f}")
print("\nClassification Report:")
print(classification_report(y_test, y_pred))

# Create visualization
plt.figure(figsize=(10, 6))
plt.scatter(X[y==0]['Points'], y[y==0], color='red', label='Loss', alpha=0.5)
plt.scatter(X[y==1]['Points'], y[y==1], color='blue', label='Win', alpha=0.5)

# Plot probability curve
X_test_sorted = np.linspace(X['Points'].min(), X['Points'].max(), 100).reshape(-1, 1)
y_test_prob = model.predict_proba(X_test_sorted)[:, 1]
plt.plot(X_test_sorted, y_test_prob, color='green', label='Win Probability')

plt.xlabel('Points Scored')
plt.ylabel('Win Probability')
plt.title('Win Probability vs Points Scored')
plt.grid(True)
plt.legend()

# Example predictions
print("\nExample Predictions:")
example_points = np.array([[95], [105], [115]])
example_probs = model.predict_proba(example_points)[:, 1]

for points, prob in zip(example_points, example_probs):
    print(f"Points Scored: {points[0]}")
    print(f"Win Probability: {prob:.3f}")
    print(f"Predicted Outcome: {'Win' if prob > 0.5 else 'Loss'}\n")
```

```
Model Coefficients:
Intercept: -26.457
Points Coefficient: 0.248

Classification Report:
              precision    recall  f1-score   support

           0       0.62      0.89      0.73         9
           1       0.86      0.55      0.67        11

    accuracy                           0.70        20
   macro avg       0.74      0.72      0.70        20
weighted avg       0.75      0.70      0.69        20


Example Predictions:
Points Scored: 95
Win Probability: 0.052
Predicted Outcome: Loss

Points Scored: 105
Win Probability: 0.397
Predicted Outcome: Loss

Points Scored: 115
Win Probability: 0.887
Predicted Outcome: Win
```

Source: own elaboration.

Interpretation

The model shows:

- Higher points totals = higher probability of win.
- The relationship between match outcomes and points totals is non-linear.
- How accurate is the classification.

There are a few ways this can be applied:

- to create performance benchmarks,
- win probability estimates and scoring targets and the game strategy that gets us to the desired outcomes.

Unit 4.3 Correlation analysis

Correlation measures the strength and direction of the relationship between two variables. It indicates **how changes in one variable correspond to changes in another. The correlation coefficient ranges from -**1 to +1, where:

- +1 indicates a perfect positive correlation (variables move in the same direction),
- -1 indicates a perfect negative correlation (variables move in opposite directions),
- 0 indicates no correlation (variables move independently).

Correlation analysis is a statistical method that examines the relationships between variables to understand patterns, dependencies, and potential causation. It helps identify which variables might be connected and how strongly.

Applications of correlation analysis include the following.

- Finance: analyzing relationships between different assets, risk assessment, and portfolio diversification.
- Healthcare: studying connections between health indicators, symptoms, and treatments.
- Marketing: understanding customer behavior patterns and product preferences.
- Sports analytics: examining relationships between player statistics and team performance.
- Environmental science: studying relationships between climate variables.
- Quality control: identifying factors affecting product quality.
- Economic analysis: understanding relationships between economic indicators.

Example

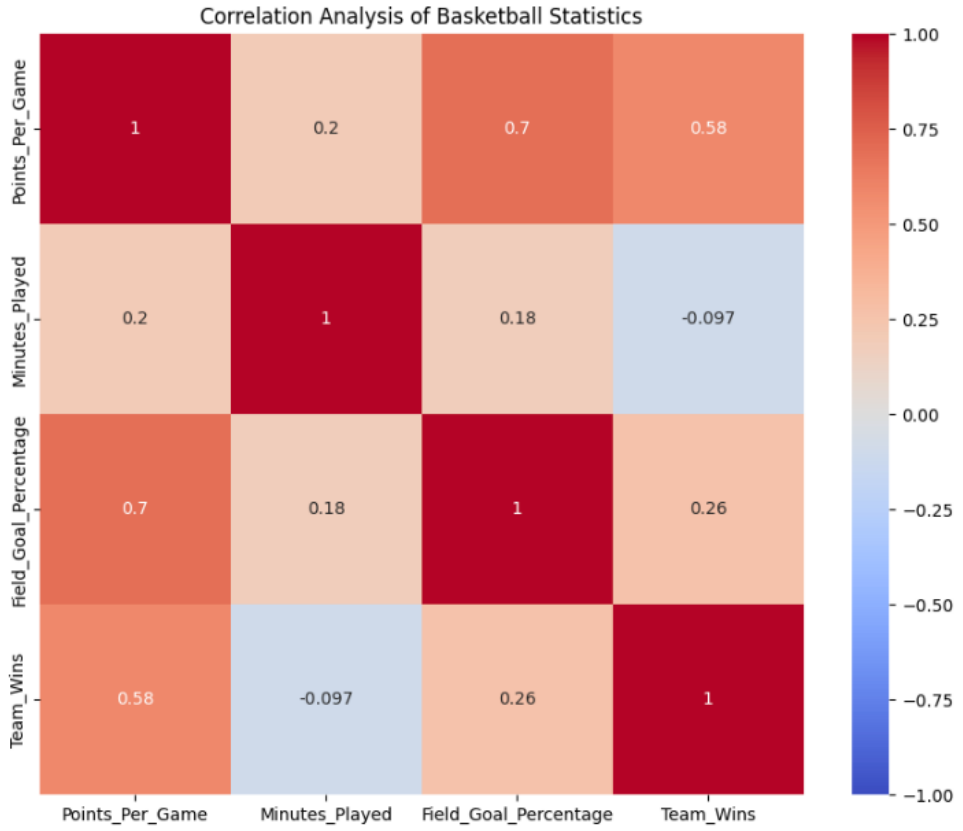
Let's use an example in sports to see how correlation analysis works.
We take some random basketball data to run correlations and map the correlations on a heatmap. Please remember that this is not actual data.

Figure 4. Correlation analysis

```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Sample NBA player statistics
data = {
    'Points_Per_Game': [28.5, 23.3, 27.1, 24.8, 26.2, 22.9, 21.5, 21.8, 25.6, 20.4],
    'Minutes_Played': [36.2, 34.8, 35.9, 33.5, 25.1, 32.4, 33.8, 31.9, 34.5, 31.2],
    'Field_Goal_Percentage': [48.5, 46.2, 47.8, 41.9, 47.2, 44.8, 45.6, 43.9, 46.8, 43.2],
    'Team_Wins': [52, 48, 40, 45, 49, 43, 26, 42, 47, 41]
}

df = pd.DataFrame(data)

# Calculate correlation matrix
correlation_matrix = df.corr()

# Create correlation heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Correlation Analysis of Basketball Statistics')

# Print specific correlations
print("\nCorrelation between Points Per Game and Team Wins:",
      df['Points_Per_Game'].corr(df['Team_Wins']).round(3))
print("Correlation between Minutes Played and Field Goal Percentage:",
      df['Minutes_Played'].corr(df['Field_Goal_Percentage']).round(3))
```

Source: own elaboration.

Figure 5. Heatmap of correlations between all the metrics in the data



Source: own elaboration.

Interpretation

- There is a strong correlation between points per game and field goal percentage. Which makes sense intuitively. The higher % of shots you make, the higher points you score.
- This is a minimal, simulated dataset, so some correlation coefficients are unrealistic. However, we will see more realistic correlation coefficients when you plug in accurate data and a more significant data sample.

4.3.1 Types of correlation coefficients

Table 1. Types of correlation coefficients

Type	Usage
Pearson correlation: this is the most used correlation coefficient.	-Measures linear relations between continuous variables. -Ranges from -1 to 1. -Assumes normal distribution.

	-Sensitive to outliers.
Spearman rank correlation	-Measures monotonic relationships – values that increase or decrease together. -Uses ranked values instead of raw data. -Doesn't assume normal distribution.
Kendall - Tau	-Based on rankings. -Best suited for small samples. -Better for hypothesis testing.
Partial correlation	-Measures the relationship between two variables while controlling for others. -Removes the effect of confounding variables. -The big drawback is that the computation is more complex than simple correlations.

Source: own elaboration.

This is not an exhaustive list; many more are used for particular scenarios.

Unit 4.4 Decision framework for selecting the right model

This section addresses deciding which model is most suitable for which data type.

Table 2. Decision based on data type and structure

Data type and structure	**Models/techniques**
Continuous data	Linear regression
Binary outcome/probability outcome	Logistic regression
Count data	Poisson regression
Time-ordered data	Time series models

Source: own elaboration.

Table 3. Decision based on the sample size

Sample size	
Small Sample (n < 30)	-Use simple models by focusing on key features.
Large sample size (n > 100)	-Start with simple models but can use more complex models.

	-Feature engineering and cross-validation become essential as the sample size gets bigger.
For medium/in-between sample sizes	-Start with simple models. -Trial and error with more complex models.

Source: own elaboration.

Table 4. Decision by features

Feature characteristics	
Few features	Standard regressions
Many features	Regularized models
Correlated features	Ridge regression

Source: own elaboration.

Table 5. Decision by analyzing performance objectives

Performance objectives	
Prediction accuracy	Ensemble methods
Inference	Statistical models
Feature selection	Lasso regression
Robust to outliers	Linear regression and spearman correlation

Source: own elaboration.

Unit 4.5 Common sports analytics questions and appropriate analysis techniques and regression models

To follow up on 4.4, we summarize the most common questions asked of sports analysts and list the recommended techniques and models for solving them. Please note that these are just recommendations, not the only way to solve the problem.

Table 6. Common sports analytics questions and appropriate analysis techniques

Type and questions	**Recommended techniques**
Player performance analysis	
How do minutes play affect scoring?	Simple linear regression Dependent variable: points scored Independent variable: minutes played
What factors predict player efficiency?	Multiple linear regression

	Dependent: efficiency rating Independent: minutes, points, rebounds, assists
Will a player score three goals in a game? Will a player exceed 20 points in a game?	Logistic regression Binary outcome: over/under three goals Features: recent performance, opponent, home/away
How is the recent form of a player	Time-series analysis
Match-up analysis	Logistic regression
Is the player overloaded? Load management	Mixed effects models
What is the career trajectory of a player?	Growth curves
What is the player's injury risk if he plays the next match?	Survival analysis combined with logistic regression to predict the probability of injury
Team performance questions	
Which factors predict winning?	Multiple linear regression Dependent: win % Independent: team statistics
How do you predict the result of a soccer match?	Logistic regression Binary outcome: win/loss Features: team stats, opponent stats, venue, etc.,
Win probability playoff odds	Logistic regression
Score prediction Win totals prediction	Poisson regression
Win margin prediction	Linear regression
Predict the final ranking of teams in the league.	Ordinal regression
Lineup analysis	
What is the ideal playing XI for a football team?	Mixed effects models
Plus/minus models	Linear regression

Identify the optimal rotation patterns for a hockey team.	Time series analysis
Optimal selection of the next play in the NFL	Multinomial regression
Find all the advantageous matches in a baseball game	Logistic regression
Valuation based problems	
How do we determine the transfer cost of a player in soccer? How to determine the optimal player salary?	Ridge regression. Dependent: salary or transfer fee. Independent: performance metrics of the player, such as age, current league, future league, etc.
How do we identify undervalued players?	Residual analysis from regression models. Comparison of predicted vs. actual performance.

Source: own elaboration.

Unit 4.6 Decision framework to select the appropriate correlation method

This section maps the most suitable correlation method for data types and problems.

Table 7. Decision framework by data type

Data type and structure	Method
Continuous	Pearson correlation
Ordinal	Spearman or Kendall-Tau
Mixed scales	Convert the data into ranks using Spearman.

Source: own elaboration.

Table 8. Decision framework by data distribution

Data distribution	Method
Normal distribution	Parametric methods
Non-Normal	Non-parametric methods
Mixed distributions	Transform or use Ranks/Spearman.

Source: own elaboration.

Table 9. Decision framework by sample size

Sample size	

Large (n > 30)	Any appropriate method
Medium (10 – 30)	Non-parametric
Small (n < 10)	Kendall's Tau

Source: own elaboration.

Table 10. Decision framework by research goals

Primary goal	
Descriptive	Basic correlation
Inference	Hypothesis testing
Prediction	Regression analysis
Classification	Discriminant analysis

Source: own elaboration.

Table 11. Individual metrics – single performance metric vs. time

Individual metrics – single performance metric vs. time	
Linear trend	Pearson
Non-linear trend	Spearman
Multiple time points	Time series
Multiple metrics	
All continuous metrics	Correlation matrix
Mixed types	Heterogeneous correlation
Hierarchical	Multilevel correlation

Source: own elaboration.

Unit 4.7 Case study: football player/team minutes analysis

4.7.1 Background

You are a data analyst at a football club tasked with analyzing minutes played by the upcoming talent, both within your club and across the league. The distribution of playing time has been a contentious topic in football, balancing between coach decisions and seasonal challenges while pursuing trophies.

4.7.2 Data collection

The analysis requires comprehensive playing time data, which can be obtained through various sources covered in earlier chapters. Since this analysis operates at the league level, you can utilize standardized league technology and data collection methods.

Data source

- Primary data source: FBref.com
- Focus: minutes played by players throughout the season.
- Scope: full-season data.

Key metrics

- MP (matches played): total number of matches participated in.
- Starts: number of matches started.
- Sub: appearances as a substitute.
- Subs: times when the player was substituted during a game.

Time available calculations

- Base calculation: available minutes per match (95 minutes).
- Example: player with 10 matches = 950 potential minutes.
- Consideration of both starting roles and substitute appearances.

Analysis approach and visualization

1. What percentage of available minutes are distributed across the squad?
2. Team rotation patterns:
    - Presence of consistent starters.
    - Coach's squad rotation preferences.
    - Distribution of 70 % of team minutes among athletes.

Calculation methodology

Minute distribution formula

- Individual player minutes/total team minutes.
- This provides insights into playing time distribution patterns.

Data visualization approach

- Focus on creating meaningful visuals that anticipate stakeholder questions.
- Emphasis on pre-emptive analysis to address potential follow-up queries.
- Goal: create comprehensive visuals that tell the complete story.

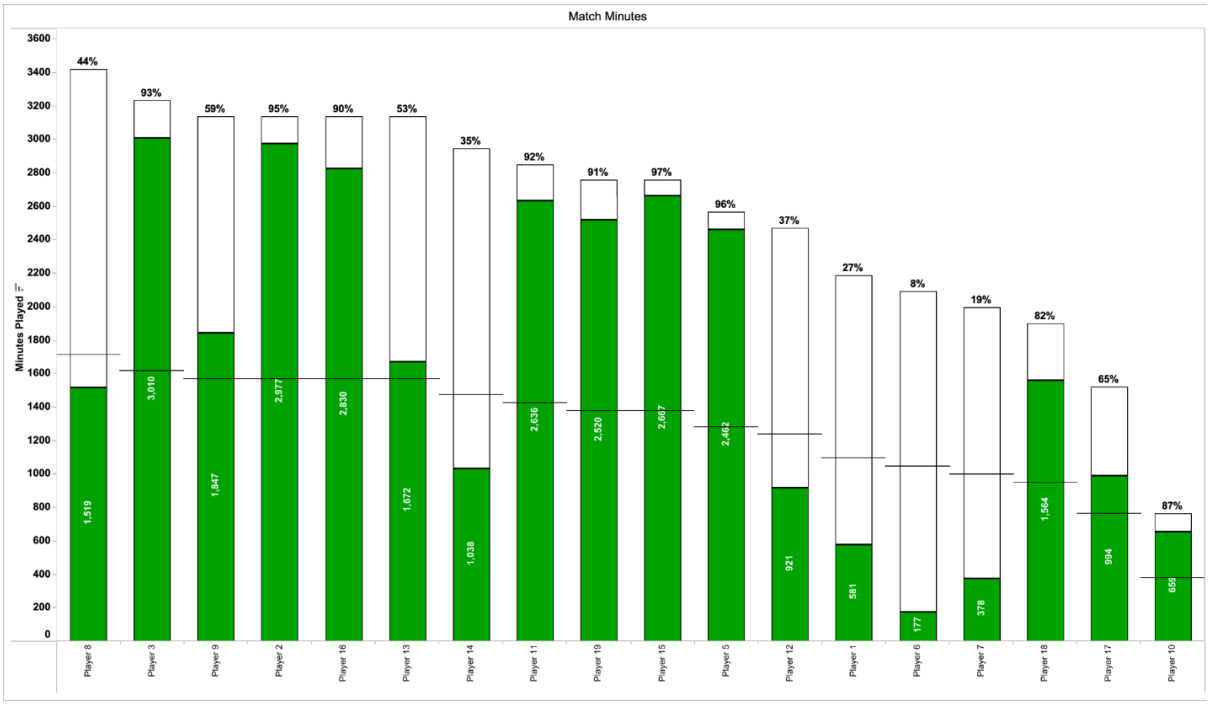
Implementation tools

1. Primary visualization tool: Tableau.
2. Data preparation steps:
    - Download data using Python.
    - Clean and organize data using previously covered Python techniques.
    - Prepare data structure for Tableau visualization.

Building the visual

1. Once you have loaded the data in the Tableau, the second step is to build a dual-axis bar chart with average lines to show if they are above or below the available minutes.

Figure 6. Dual-axis bar chart



Source: own elaboration.

2. Using the Tableau calculation, you can adjust the findings to fit your questions. As discussed above, you must write a calculation to answer each question and add it to the Dashboard.

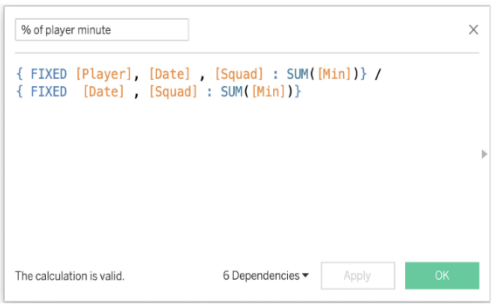
Figure 7. Tableau calculation

Calculation: 1

Calculation: 2

Calculation: 3

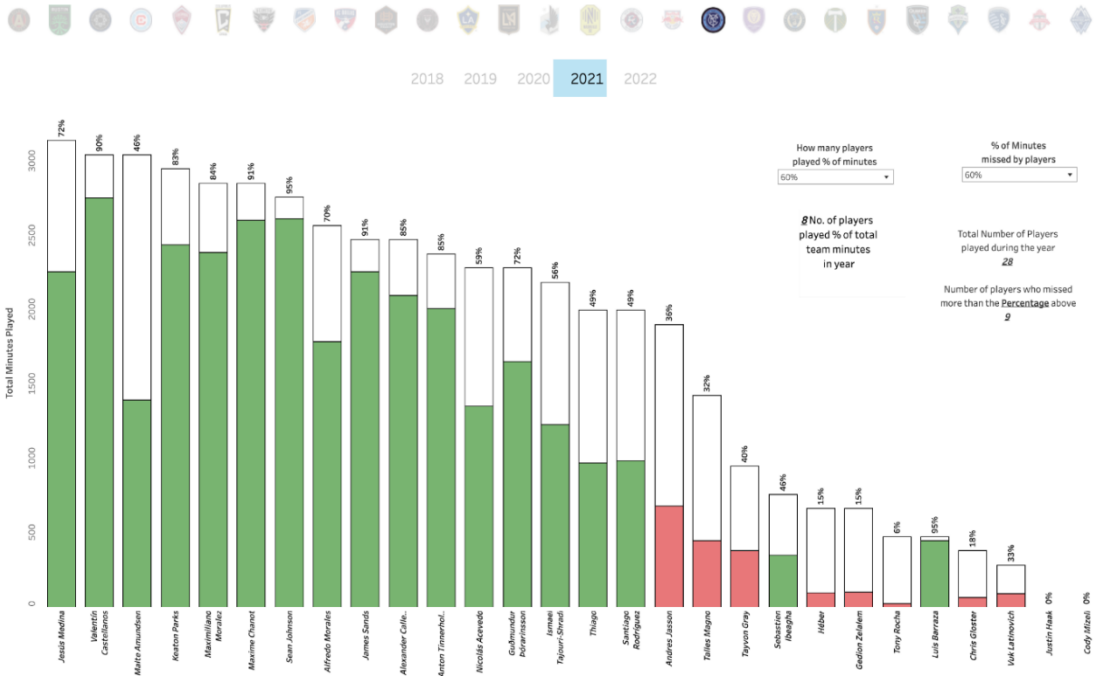
Calculation: 4



Source: own elaboration.

3. Add more detail about navigating between teams, years, and stats; you will have a complete dashboard to present to the coaches.

Figure 8. Complete dashboard



Source: own elaboration.

This case study demonstrates the importance of thorough data analysis and effective visualization in football analytics, particularly in understanding player utilization patterns. Advanced calculations can help you learn many hidden gems and the power of data visualization.

References

Casella, G., and Berger, R. L. (2002). *Statistical Inference*. Cengage Learning

Fox, J. (2015). *Applied Regression Analysis*. SAGE.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer.

James, G., Witten, D., Hastie, T., Tibshirani, R., and Taylor, R. (2013). *An Introduction to Statistical Learning*. Springer.

Kutner, M. H., et al. (2004). *Applied Linear Statistical Models*. McGraw Hill/Irwin.

Lehmann, E. L., and Romano, J. P. (2005). *Testing Statistical Hypotheses*. Springer.

McKinney, W. (2017). *Python for Data Analysis*. O'Reilly.

Wasserman, L. (2004). *All of Statistics*. Springer.

Wickham, H., and Grolemund, G. (2016). *R for Data Science*. O'Reilly.