Development of a Human Resources Data Visualization Strategy for Sport Lisboa e Benfica – with an Emphasis on Compensation and Benefits

NILS VAN DER KEMP

Abstract

Visualizing and analyzing data are ubiquitous methods for efficient decision-making in companies today. Addressing an insufficiently structured reporting system at Sport Lisboa e Benfica, this thesis aims to develop an innovative data visualization strategy for their human resources department. To assist the company in decision-making, five Power BI dashboards were created using synthetic employee data. These dashboards incorporate a holistic view from an external perspective, supplemented by suggestions drawn from industry best practices. The resulting dashboard strategy demonstrates that problems and correlations can be quickly identified, emphasizing the efficiency and effectiveness of data-driven decision-making.

Keywords:

Data Visualization, HR Analytics, Sports Management, Data Randomization

This work used infrastructure and resources funded by Fundação para a Ciência e a Tecnologia (UID/ECO/00124/2013, UID/ECO/00124/2019 and Social Sciences DataLab, Project 22209), POR Lisboa (LISBOA-01-0145-FEDER-007722 and Social Sciences DataLab, Project 22209) and POR Norte (Social Sciences DataLab, Project 22209).

Group Part

1. Methodology

1.1 Data Source and Analysis Tools

To implement an efficient data visualization strategy for Sport Lisboa e Benfica and thus solve an elementary problem underlying this thesis, a data source was needed first and foremost. As the data to be visualized contains highly sensitive personal information, i.e., specific salary details, the organization was unable to provide access to their actual data. Consequently, all the information included in the subsequent visualizations is based on educated assumptions. None of the values therefore reflect the reality at Sport Lisboa e Benfica in any way but solely serve as a basis for developing a KPI visualization strategy. Given that the objective of this work is to develop a strategic approach rather than to analyze actual values, the absence of data does not affect the quality of the results. Nevertheless, it was tried to re-design the data as realistically as possible.

The artificial generation of the missing personal data was accomplished using Python – an interpreted, high-level programming language, known for its clear syntax and computational efficiency (Reshma Ramdas 2019, 71-72). For the presentation of data, Microsoft Power BI was applied. Power BI can be defined as a "unified, scalable platform for self-service and enterprise business intelligence" (Microsoft n.d.) which allows users to visualize data from a variety of different sources. In other words, this was the application used to process the synthesized Python data and ultimately implement the KPI visualization strategy by building different dashboards.

1.2 Approach to Data Preparation

1.2.1 Employee Master Data

To be able to provide all relevant data, three different datasets were created, one with employee

master data, followed by employee history and general applicant data. In the following, the creation of these datasets will be described in more detail, supported by some exemplary code snippets for improved clarification. For an even more comprehensive understanding of the data generation process, the full code can be found in the appendix.

In the first step, some additional libraries were imported to expand the functional possibilities of Python. The most important extension during the following data generation was the Pandas library. Pandas is one of the most popular and frequently used libraries, which is particularly suitable for cleaning, manipulating, and transforming data (Gholizadeh 2022, 142). Another widespread library used here is NumPy. In combination with the Random module, it was specifically required for the random distribution of values. In addition, functions from the datetime module were imported for subsequent time-related operations, and the display of warning filters was disabled. This first step of the code generation can be seen in the following screenshot.

```
import pandas as pd
import numpy as np
import random as rd
from datetime import datetime, timedelta
import warnings
warnings.filterwarnings("ignore")
```

Figure 1: Import of relevant libraries

Using the Pandas library, the initial data frame has been established to map the employee master data. The dataset was to comprise 20,000 entries, with each entry representing the information of one individual employee. Furthermore, it was assumed that there are 750 people currently employed at SL Benfica. To store the individual master data in the dataset, several columns had to be added, which were then filled with predefined entries along with their frequency distribution. The following excerpt shows an example of three different columns. The “gender” column, for instance, was filled with “male” in 70% of the cases and with “female” in 30% of

the cases.

```
employees = 20000
active_employees = 750
df = pd.DataFrame()

df["employee_id"] = df.index + 1
df["corporation"] = np.random.choice(["corporate","sport"], employees, p=[0.4, 0.6])
df["gender"] = np.random.choice(["male","female"], employees, p=[0.7, 0.3])
df["age"] = np.random.choice(["under 25","25-35","35-50","50-66","over 66"], employees, p=[0.18, 0.31, 0.27, 0.15, 0.09])
```

Figure 2: Initialization and first column creation

Some columns, on the other hand, could not be filled arbitrarily but had to be derived from others to ensure a certain logic in the data. The department of an employee, for example, should be based on the "corporation" column to ensure that no sports-related jobs are displayed if someone is a corporate employee. Likewise, the tenure of an employee is based on the "age" column, as an employee cannot be with the company longer than his or her age. For this study, it was assumed that a person was hired at the age of 18 at the earliest. The coded implementation of these cases can be seen in the following screenshot.

```
df["department"] = np.where(df["corporation"] == "corporate", np.random.choice(["HR", "Marketing & Sales",
                            "Controlling", "Finance & Administration", "IT", "Media & Contents"], employees),
                            np.random.choice(["Scouting", "Youth Football", "Professional Football"], employees))

def generate_random_tenure(age_interval_end):
    upper_bound = (age_interval_end*12)-(18*12)
    return rd.randint(1, upper_bound)

df['tenure_in_months'] = df['age_interval_end'].apply(generate_random_tenure)
```

Figure 3: Column filling with dependent variables

As mentioned at the beginning of this chapter, the dataset comprises 20,000 rows, of which only 750 represent active employees. For the 19,250 inactive employees, the direction of termination was also recorded, indicating whether they were dismissed by SL Benfica or left the organization voluntarily. For the latter cases, a termination reason was similarly assigned, which was again selected at random from predefined entries for each employee. This data was intended to represent the responses of churned employees given in their exit survey and provided a basis for assessing employee attrition at a later stage.

As a foundation for compiling the employee history, an entry date and, in the case of inactive employees, an exit date has been determined. For active employees, the start date was calculated

by subtracting the tenure in months from the current day, while the end date was left blank. For inactive employees, however, a period was defined which was bounded by 01 January 1963 as the earliest possible start date and the current day minus the tenure in months as the latest start date. Within this period, a date was generated randomly. To obtain the exit date, the tenure in months was added to the start date just defined. The respective lines of code can be observed below.

```
def calculate_start_date(row):
    tenure_in_months = row['tenure_in_months']
    active_employee = row['active_employee']
    if active_employee == 1:
        start_date = datetime.now() - timedelta(days=tenure_in_months * 30)  # Approximating 30 days per month
    else:
        start_date_min = datetime(1963, 1, 1)
        start_date_max = datetime.now() - timedelta(days=tenure_in_months * 30)
        if start_date_min < start_date_max:
            start_date = start_date_min + timedelta(days=rd.randint(0, (start_date_max - start_date_min).days))
        else:
            start_date = start_date_min
    return start_date.date()

df['start'] = df.apply(calculate_start_date, axis=1)
df["end"] = df.apply(lambda row: pd.to_datetime(row["start"]) + timedelta(days=row["tenure_in_years"] * 365)
                        if row["active_employee"] == 0 else None, axis=1)
```

Figure 4: Definition of employee's entry and exit dates

As a major focus of this work is on employee benefits, this aspect had to be included as well. Therefore, a column was added to the dataset for each benefit specifying whether an employee is entitled to that particular benefit. Information on the individual benefits that Sport Lisboa e Benfica offers its employees was not available in the context of this thesis, which is why five arbitrary, albeit realistic, benefits were chosen. These specific benefits will be highlighted in more detail in chapter 4.2, when the visualization strategy for benefit KPIs will be described. The allocation of benefit entitlements to employees again followed a randomized procedure.

The final step in creating the master data table was the inclusion of metrics on employee satisfaction in the company. For this purpose, it was assumed that an internal company survey is conducted at regular intervals, for example, once a year, where employees provide the company with feedback on various aspects. To illustrate the results of this hypothetical survey, five additional columns were added to the dataset, each representing one evaluation criterion.

Finally, each of these columns was randomly filled with integer values between 1 and 5, with 5 representing the best and 1 the worst possible evaluation of a criterion.

1.2.2 Historical Employee Data

Now that master data for 20,000 employees has been available, historical information for each employee should also be captured, i.e., data that may change over time, such as salary payments. To achieve this, certain master data had to be duplicated for each employee. In line with the theory of database normalization, this required the generation of a separate dataset to reduce redundancies, improve the integrity of the data, and enable more efficient queries (Eessaar 2016, 9). Each employee should have an entry for each year in which he or she was employed. To do so, a copy of the master data table was created, and the data of each employee was duplicated by the tenure in years. In a new column “year”, every year of tenure should now be indicated for each employee, starting with the year in which an employee joined the company. Each row thus represents a year in which an employee was active. The utilized code can be seen below.

```
df_hist = df
duplicated_rows = []

for _, row in df_hist.iterrows():
        duplicated_rows.extend([row] * (row["tenure_in_years"] + 1))

df_hist = pd.DataFrame(duplicated_rows, columns=df_hist.columns).reset_index(drop=True)

df_hist["year"] = df_hist["start"]

def increment_years(group):
    group["year"] += np.arange(len(group))
    return group

df_hist = df_hist.groupby("employee_id").apply(increment_years)
df_hist = df_hist.reset_index(drop=True)
```

Figure 5: Duplication of master data and year incrementation

At this point, the newly generated data table comprised around half a million entries. In the next step, the data should be duplicated once again so that it can be presented not only on a yearly but also on a monthly basis. Therefore, each row had to be duplicated twelve times, which would have resulted in a dataset of around 6 million entries. To avoid such large amounts of

information, all data before 2005 had been removed before duplicating. On the one hand, this enabled a focus on more up-to-date information and, on the other hand, reduced processing time for subsequent queries and operations. As with duplication by year, a new column has also been created for the months, which counts from 1 to 12 for each combination of employee ID and year.

Since the dataset was supposed to represent employee data only for the duration of their active employment, showing exactly twelve months for each year of an employee's tenure did not correspond to a realistic scenario. This would mean that an employee always started their contract in January while an inactive employee always left in December. To keep the former issue more realistic, a random number of rows between 0 and 11 was deleted from the dataset for each employee's first year. The same logic was applied to all inactive employees to possibly remove entries from their last year, illustrated in the following screenshot.

```
inactive_rows = df_hist[df_hist['start'] == df_hist['year']].copy()
inactive_rows.sort_values(['employee_id', 'year', 'month_number'], ascending=[True, True, True], inplace=True)
inactive_rows['drop'] = 0  # Initialize the drop column

for _, group in inactive_rows.groupby('employee_id'):
    num_ones = rd.randint(0, 11)
    if num_ones > 0:
        last_rows = group.head(num_ones).index
        inactive_rows.loc[last_rows, 'drop'] = 1

df_hist['drop'] = np.where(df_hist['active_employee'] == 1, 0, inactive_rows['drop'].reindex(df_hist.index, fill_value=0))
df_hist = df_hist[df_hist["drop"] == 0]
```

Figure 6: Adjustment of the termination month of inactive employees

With the preliminary dataset structure now in place, it could finally be expanded by new information, starting with monetary data in terms of salary and benefit payments. The amount of salary paid to an employee depends on a variety of factors and is therefore difficult to represent perfectly. However, this was not a limiting factor, as this work is merely about developing a visualization strategy. For simplicity, four salary ranges were generated, from which a random value was drawn depending on the management level of an employee. For a rough classification of the Portuguese salary level, the information provided by PORDATA (n.d.) was taken as a benchmark. The defined ranges stretched from €4200 to €6000 gross

monthly income for a top manager to €1600 to €2300 for a support employee.

The costs incurred by SLB for the respective benefits were set arbitrarily. This way, the benefit of flexible working hours, for example, was estimated to incur costs of €50 if an employee is entitled to receive this benefit and €0 otherwise. To keep it as uncomplicated as possible in this case too, it was assumed that the costs associated with each benefit have not changed throughout the organization's history.

However, as salaries do not usually remain constant over the entire duration of an employee's contract, an additional component was added to the data, namely promotions. A new column has been created to reflect whether an employee has been promoted each month. To maintain a reasonably realistic balance in the number of promotions without completely randomizing them, it was defined that an employee can be promoted no more than once every four years on average. This logic, when converted into code, can be taken from the following excerpt.

```
df_hist["promotion"] = 0
promotions_counter = {}

for index, row in df_hist.iterrows():
    employee_id = row["employee_id"]
    if promotions_counter.get(employee_id, 0) < (row["tenure_in_years"] / 4):
        promotion_value = np.random.choice([0, 1], p=[0.7, 0.3])
        df_hist.at[index, 'promotion'] = promotion_value
        promotions_counter[employee_id] = promotions_counter.get(employee_id, 0) + promotion_value

df_hist['promotion'] = df_hist['promotion'].astype(int)
```

Figure 7: Definition of the frequency of promotions

Based on this newly established information, two further columns have been created. The first shows the absolute salary increase if an employee has been promoted, while the second puts this increase in relation to the previous salary and thus expresses the percentual salary increase. The extent of a salary increase followed a uniform distribution within the interval of 1% and 10% of an employee's base salary. Finally, the updated salary, i.e., base salary plus salary increase, was displayed in an additional column.

Following the expansion of the dataset to include monetary data, the next step was to incorporate time-related parameters, like regular working hours, vacation times, sick days, or

overtime. The regular working hours were determined based on an employee's contract type. 160 hours per month were assumed for full-time workers and interns, 80 for part-time workers, and a random value between 8 and 160 for freelancers. Moreover, full-time employees were granted the right to an annual 30 vacation days, which corresponds to 12.5% of their regular working hours. This percentage was also used to determine the entitlement of vacation days for the remaining contract types. The resulting values were rounded down to integers. For example, a freelancer who worked 100 days in a specific year would be granted 12 vacation days for that year. Sickness-related absences were determined randomly again. Also, it was assumed that an employee was absent for a maximum of 30% of their net working time (regular working time minus vacation). Similarly, the number of overtime hours was specified, yet in this case with a maximum of 18% of the regular working hours, it was assumed that employees are absent for a greater duration than they work extra hours.

At this point, the dataset still contained all the information duplicated from the master data table, as part of it served as a reference for creating new columns. However, since it was no longer needed, most of this data could be removed. The only exception was to keep the employee IDs, as they represented the foreign key to establish a connection to the master dataset. A foreign key is a reference to the primary key of another table – the parent table. The datasets are linked by this reference so that data from the other table can be easily retrieved (Kraleva et al. 2018, 121; W3Schools n.d.).

Finally, to complete the employee history dataset, the columns "starter" and "leaver" were added. The former shows a 1 for the first entry of an employee – sorted by years and months – and a 0 for all remaining rows. The latter indicates the same but for the last row and only for inactive employees. This facilitated subsequent cumulative operations, such as displaying the number of employees who started each year, achieved by drawing the total of the column "starter", grouped by the "year" column.

1.2.3 Applicant Data

The third and last dataset created encompasses applicant data. This includes all active and inactive employees generated previously, as well as individuals who have applied to Sport Lisboa e Benfica at some point but were never hired. The data collected within this table was essential for the illustration of all recruiting key figures.

First, a dataset consisting of one column and 50,000 rows was created. In this column, the 20,000 employee IDs of previously created employees were recorded, the remaining 30,000 rows were filled with 0 and were intended to represent those applicants who were not hired. Using the employee ID as a foreign key, some information from the master data table, such as an employee's gender, age, or department, could then be added to the new dataset via a left join. The left join ensures that all data in the left table, in this case, the new table, is retained, regardless of whether there is matching data in the right table – in this case, the master data table (Bhanuprakash, Jayaram and Nijagunarya 2014, 22). Consequently, no master data was available for those rows that were filled with the employee ID 0, as this ID did not exist in the master data table. For these 30,000 entries, new data still had to be developed. To achieve this, the same logic as for the master dataset was applied, as can be seen below, on the example of an employee's educational level.

```
def edu(row):
    if row["employee_id"] == 0:
        return np.random.choice(["High School", "Bachelor's Degree", "Master's Degree", "PhD"], p=[0.3, 0.4, 0.2, 0.1])
    else:
        return row["educational_level"]

df_app["educational_level"] = df_app.apply(edu, axis=1)
```

Figure 8: Generation of applicant master data

Furthermore, the recruitment dashboard – which will be presented in a later chapter – resulting from the applicant data was to contain information on the applicant's origin, in other words how an applicant became aware of a particular position at SL Benfica, for example through job portals. For this purpose, Python was again given a selection list from which it randomly drew

entries and stored them in the additionally generated “source” column.

To map the application process accurately, separate columns have been added for each step that an applicant has to go through. These columns were filled with 1 if an applicant has gone through this step and with 0 if not. All applicants with an employee ID different from 0, i.e., all those who were ultimately hired, had passed every stage of the application process before. For applicants who were not hired, the column for the last step before hiring has now been randomly filled with 0 or 1. In this way, the application process was gone through in reverse, which means that everyone who made it to this penultimate step also must have gone through all the other stages beforehand. For those who did not make it to the penultimate step, a 1 or 0 was randomly assigned for the stage before that, and so on. Below the programmatic implementation of the third step can be seen as an example.

```
def step3(row):
    if row["trial_work"] == 1:
        return 1
    else:
        return rd.randint(0,1)

df_app["interview"] = df_app.apply(step3, axis=1)
```

Figure 9: Construction of the application process

Beyond this information, the case that an applicant could have been hired, but rejected the contract offer on their initiative, should also be depicted. The probability of this scenario occurring was assumed to be 27%.

The final challenge was to calculate the duration of the application process for each applicant and the associated costs. The assessment of the duration was again following a uniform distribution, with differently sized intervals per step. For instance, a value between 1 and 17 days was selected to indicate the days passed between the completion of the last step before recruitment and the actual declaration of the contract. This information also allowed for determining the time to hire and time to fill for each employee, which will be discussed in more detail later when presenting the recruiting KPIs. The costs incurred for each application step were split into fixed costs and variable costs. The variable costs of a step were calculated

depending on the duration. For the sake of simplicity, a daily rate of €10 was assumed, i.e., if a step took 10 days, the corresponding costs would amount to €100. The fixed costs, however, varied for each step. For instance, it was assumed that one day of trial work caused costs of €230 per applicant. This cost data enabled the determination of the cost to hire, which will also be highlighted in a later chapter. The exact time and cost distribution can be obtained from the following snippet.

```
df_app["time_hired"] = df_app.apply(lambda x: np.random.randint(1, 17) if x["hired"] == 1 else 0, axis=1)
df_app["time_trial_work"] = df_app.apply(lambda x: np.random.randint(1, 19) if x["trial_work"] == 1 else 0, axis=1)
df_app["time_interview"] = df_app.apply(lambda x: np.random.randint(1, 20) if x["interview"] == 1 else 0, axis=1)
df_app["time_screening"] = df_app.apply(lambda x: np.random.randint(1, 26) if x["screening"] == 1 else 0, axis=1)
df_app["time_application"] = df_app.apply(lambda x: np.random.randint(7, 26), axis=1)

df_app["cost_hired"] = df_app.apply(lambda x: 1200 + x["time_hired"]*10 if x["hired"] == 1 else 0, axis=1)
df_app["cost_trial_work"] = df_app.apply(lambda x: 230 + x["time_trial_work"]*10 if x["trial_work"] == 1 else 0, axis=1)
df_app["cost_interview"] = df_app.apply(lambda x: 160 + x["time_interview"]*10 if x["interview"] == 1 else 0, axis=1)
df_app["cost_screening"] = df_app.apply(lambda x: 70 + x["time_screening"]*10 if x["screening"] == 1 else 0, axis=1)
```

Figure 10: Time and cost factors of the application process

As a result, the third dataset was completed as well, so that all the master data as well as the employee history and applicant data could be saved in a CSV file and imported into Microsoft Power BI, ready for visualization.

1.3 Dashboard 5: Fluctuation

The final dashboard focuses on employee fluctuation and specifically compares employees who are still active with those who are no longer employed by Sport Lisboa e Benfica. These groups have been differentiated by color for the entire dashboard. All data shown in gray refers to retention, i.e., those employees who are still active. The data in red or white refers to attrition, and therefore to employees who have already left the company. This includes both churners who have left voluntarily and layoffs by the company. A legend that reflects this color division has been placed directly underneath the dashboard title. Below this, a slicer has been added, which by ticking the box only displays attrition data and thus hides everything displayed in gray. Furthermore, the usual four slicers for filtering gender, department, organization, and management level have been integrated again.

The first visualization accurately demonstrates the results of the employee satisfaction survey. All scores were displayed separately for the two groups, retention, and attrition. To present the results in this format, numerous visuals were combined. For the sake of simplicity, the structure will now be explained using the first evaluation criterion, which was the potential career opportunities. On the left-hand side, there are two single-row matrix visuals, one above the other, which show the distribution of evaluation points for this criterion for each of the two groups. To stick to the data shown in the appendix, 244 employees rated the career opportunities at SLB with 5 out of 5 within the retention group, whereas only 31 did so within the attrition group. Moreover, a color coding was added to the rows of the matrix visuals, according to which a score was colored in a darker shade the more votes it received. This conditional formatting was the reason why the rows were all displayed in separate matrix visuals, as it was the most convenient way to color them – according to the color scheme – in shades of gray for retention or in shades of red for attrition. In addition, two card visuals have been added on the right-hand side next to the criterion, which shows the average rating scores of retained employees and lost employees. In this way, the presentation of each evaluation criterion was structured, from career opportunities down to work-life balance.

The next graph is a clustered column chart again, which shows the development of attrition in the current year, over the individual months. To the right of the column chart, this attrition can be examined more holistically. Firstly, the total loss of staff for the entire year is shown, which is calculated from the sum of the individual columns. Secondly, a year-on-year comparison is drawn, both in absolute and relative terms. For the imaginary data behind the dashboard, this would mean that in 2023 with a total of 517 employees, SL Benfica has lost 62 fewer employees than in the previous year, which means that attrition fell by 10.71%. Like the benefits dashboard, an arrow has been added here to help interpret the results. As a reduced loss of staff is usually something positive, a green arrow pointing downwards is shown here. In the event of

increased attrition compared to the previous year, a red arrow pointing upwards would be shown. If attrition remained the same, it would disappear completely.

On the top right of the dashboard, a clustered bar chart can be seen. This bar chart addresses the reasons why employees left the organization voluntarily. It is the only visual on the fluctuation dashboard that does not show the attrition but explicitly focuses on the churners. By default, the reasons for termination are obtained from SLB via an exit survey, carried out when an employee leaves the company. In the bar chart, the individual reasons have now been ranked according to the frequency of their indication and sorted by size. The data labels mark the relative frequencies for each termination reason. Finally, the slider on the far right of the visual can be utilized to scroll to the bottom.

Underneath this bar chart, five different donut charts are displayed, one for each age group. Again, a distinction was made between retention and attrition within these donuts, with the gray slices reflecting the retained and the white ones the lost employees. Here the slices have been labeled with the absolute number of employees again.

At the bottom, two final visuals have been integrated, a stacked column chart and a clustered bar chart. The column chart shows the fluctuation rate for different groups, measured by their tenure at SLB. Like the comparison of overtime and absent hours in the performance dashboard, the columns have been stacked in such a way that attrition increases towards the bottom and retention increases towards the top. The data labels displayed mark the difference between retention and attrition for each tenure group. The clustered bar chart at the bottom right now provides a final comparison of the data for retention and attrition depending on the educational degree achieved, which completes this last dashboard.

1.3.1 Dashboard 5: Fluctuation

The remaining dashboard to be analyzed concerns the topic of fluctuation. This dashboard is

designed to help Sport Lisboa e Benfica identify possible causes and patterns in employee exits to tackle them in the future.

The results of the employee satisfaction survey provide a first indication here. From these combined matrix and card visuals, the company can easily see how general satisfaction with the individual assessment criteria has been rated. Especially because the data for each point was divided into retained and lost employees, it is possible to estimate the extent to which a criterion may have been responsible for the exit of employees. As already mentioned in the chapter describing this dashboard, attrition includes both churners and layoffs. This means that if, for example, compensation was rated particularly poorly among the lost employees, it does not necessarily mean that it led to many employees quitting. Equally, this attrition could also include many company layoffs because the employees' dissatisfaction might have hurt their performance. These are just sample cases for interpreting this visualization. However, both cases would show that dissatisfaction hurts employee retention.

The following visual provides a profound overview of the development of staff losses in the current year, as well as a comparison to the previous year. This information is particularly relevant concerning recruitment decisions. For instance, if the company is pursuing a growth strategy and finds that 50 employees were lost in the previous month, it will now have to hire at least 50 new employees – without considering the staff that has already been recruited for the next month. If the conversion rate of 8.1% from the recruitment funnel is considered, this scenario would result in around 617 applications that would have to be generated to recruit 50 employees. Furthermore, any seasonal staff losses can be recognized from this column chart, which ensures a certain degree of planning security for the company, particularly in the case of recurring patterns.

The bar chart showing the termination reasons indicated by churners in their exit survey is a very powerful way of highlighting the company's perceived problem areas. Some of the

responses provide the company with fewer insights, while others are extremely meaningful. For example, if employees quit for personal reasons or are only looking for a new professional challenge, it is not possible to identify any direct fault on the part of the company. However, if employees terminate their employment for reasons such as salary package or work schedule, it is something that the organization can influence directly. If a particularly large number of employees leave due to dissatisfaction with the work schedule, the organization might think about hiring more employees to better distribute the tasks and reduce the workload for employees in general.

Using the five donut charts, HR managers can also monitor how the churn rate is reflected in the demographic context. If, for instance, a particularly high churn rate was to be identified among the two younger age groups, it might be an indication of the corporate culture possibly no longer being contemporary. When analyzing the oldest group, it should be noted that exits due to retirement are also included here, which may slightly distort the data visualized in this donut chart.

Finally, at the bottom of the dashboard, employee fluctuation was examined in terms of tenure and education level. The former shows briefly how many years people tend to be employed in the company before they leave. By additionally indicating the gray columns for retention, the exits can also be easily measured against the general headcount within a group. This visual again allows measures to be taken to reduce any noticeably high churn rates within subgroups of the total staff. The bar chart in the bottom right corner now compares the reliability of employees according to their educational background. For example, people with a high academic degree may have extremely high churn rates because the development opportunities in the company do not appeal to them or because they have higher salary expectations.

All in all, it can be said that each dashboard can be of great benefit to SLB, especially by going into more detail and drilling down into the data, for example by focusing on only one

department and management level. However, the dashboards are particularly beneficial in combination with each other, as already demonstrated in parts of this discussion section. Just as in a company in practice, the individual aspects covered by the dashboards are also interconnected and provide a great help in decision-making, especially when looking at the big picture.

2. Conclusion

This project aimed to develop an efficient data visualization strategy for Sport Lisboa e Benfica that sets innovative, creative incentives and ultimately simplifies the company's decision-making process in HR management. Backed by an extensive literature study and an analysis of various best practices in HR analytics, five Power BI dashboards were created from artificially generated data.

Based on the results, it can be concluded that a strategically well-elaborated data visualization in the HR area can yield enormous value and thus underlines the crucial role of HR analytics.

Concerning employee compensation, the dashboard created was not only able to provide a precise overview of salary structures but also gave indications on many levels as to how the salary package offered is perceived by employees. A salary that is considered too low usually results in dissatisfaction. This work has shown that this scenario can be identified more quickly by visualizing certain KPIs and can therefore be combated more efficiently.

The same applies to the provision of employee benefits. The visual presentation of these KPIs helps SLB to keep an eye on the benefit-cost structure and to recognize patterns between these payments and employee satisfaction as well as the associated loyalty.

Visualizing recruiting data has shown that it can provide a company with a certain degree of planning certainty about the available human capital on the one hand and, on the other, with enormous assistance in optimizing the recruitment process. The time- and cost-related

recruiting KPIs presented provide the organization with a profound basis for developing a hiring strategy.

When examining the performance dashboard, it became clear that important insights can also be gained from the hours worked by employees, particularly concerning absenteeism. Conspicuously high absenteeism might again be a sign of dissatisfaction and a lack of motivation but can also indicate health problems. The KPIs were developed to provide more precise information about this cause and thus help to introduce preventative measures.

Illustrating fluctuation metrics supports SLB in identifying possible causes and patterns in employee exits. In this context, employee satisfaction once again plays a decisive role. The visuals help to uncover connections between dissatisfaction and attrition and to take corrective actions to solve these issues in the future.

Despite the major limitation of the absence of real data, it can be concluded that each of the visual representations of HR data can be of great benefit to SL Benfica. Zooming into the data using the built-in filters and the cross-functional, dashboard-overarching use of the various visuals with each other represents an extremely useful strategy and offers great potential for improving decision-making.

Individual Part – Nils van der Kemp (55612)

3. Contextualization and Literature Review

3.1 The Role of HR in Sports Organizations

According to Armstrong (2008, 5), human resources are the most precious component within an organization. The rationale behind the statement is straightforward: every organization needs human individuals to collaborate and help the organization achieve its defined objectives. Now, HR assumes a crucial function in attracting, employing, and maintaining these individuals (Oladapo 2014, 20). In this context, it is not only about individuals who are experienced and

talented but also about the alignment on a personal level and the representation of the company's culture and values. Achieving this is vital for an organization to gain a competitive edge and is thus regarded as crucial for economic success (Price 2023; Aslam et al. 2014, 88). Just as this applies to all organizational forms, it also holds for sports organizations. Especially in the context of growing commercialization in the sports industry, the business aspect within sports organizations has significantly gained importance and will most likely continue to do so. Correspondingly, the significance of HR increases, implying that sports organizations must progressively invest in this domain to ensure organizing operations efficiently (Weerakoon 2016, 15-16).

The distinctive feature in sports organizations is that human resource management takes place on two separate levels, namely, on-field and off-field. The former encompasses all individuals directly involved in their respective sport, such as players and coaches. The latter, on the other hand, deals with the staff handling business-related and administrative tasks (Covell et al. 2012, 313-315). While some argue that sports organizations primarily focus on off-field human resources (ibid.), other sources suggest that the emphasis of HR lies in the on-field area (Gheorghe 2015, 150). In practice, an organization's success is likely contingent upon the effective synergy between its on-field and off-field operations, with both the performance on the field and the execution of business tasks playing integral roles.

3.2 The Importance of Compensation and Benefits

Being one of the main components of an organization, human resource management is in turn also divided into different sub-areas. Alongside the recruitment process, this encompasses elements such as performance management, training, and development, as well as the provision of compensations and benefits (Armstrong 2008, 40; van Vulpen n.d.).

Compensation usually refers to the comprehensive financial rewards provided to employees in

exchange for their work, including salaries, bonuses, profit-sharing, pension plans, and paid leave. In contrast, benefits represent voluntary, non-financial forms of remuneration offered to employees in addition to their regular wages. Among these are items such as health insurance, retirement benefits, daycare, family leave, or a flexible working schedule (Omar 2019, 112; Bester 2023).

Compensation and benefits play a critical role in business practice and human resource management (HRM). Not only are these components the largest expense factor for an organization (Biswas 2013, XXV; Enderes 2023), but they are also closely linked to job satisfaction and employee retention (Shtembari, Kufo and Haxhinasto 2022, 1). According to Shtembari, Kufo and Haxhinasto (2022, 1), a survey conducted by the Society for Human Resource Management (SHRM) in 2018 revealed that 92% of employees considered compensation and benefits critical to their job satisfaction. It has become evident that these factors have a significant impact on an employee's decision whether to look for another job or stay with the current employer. A remarkable proportion of 29% said that their company loyalty is closely tied to the compensation and benefits they receive (ibid.).

However, the importance of compensation and benefits extends across the entire employee lifecycle, from recruitment to long-term retention. Therefore, the challenge in human resource management lies in not only attracting qualified workers but also in maintaining the satisfaction of existing employees, thus ensuring their commitment to the organization. Beyond retaining employees, however, compensation and benefits often yield another positive effect – improved performance. Numerous studies agree that satisfied and highly motivated employees tend to be more efficient and productive (Candradewi and Manuati Dewi 2019, 136; Aslam et al. 2014, 88). Consequently, the effective design and implementation of compensation and benefits packages can and already have been shown to influence an organization's long-term success significantly (Aslam et al. 2014, 87-88; Shtembari, Kufo and Haxhinasto 2022, 1-3).

3.3 The Importance of HR Analytics in Modern Organizations

Following the general overview of HR in sports organizations and emphasizing the important role of compensation and benefits, it is essential to shed light on another key aspect of modern human resource management – HR analytics.

Van den Heuvel and Bondarouk (2017, 130) define HR analytics as "the systematic identification and quantification of the people-drivers of business outcomes, with the purpose of making better decisions". Specifically, it comprises the methodology of collecting, examining, and displaying HR data to evaluate the impact of specific HR metrics on organizational performance, thereby deriving actionable recommendations (Manikandan and Wilson 2023, 3091).

HR analytics has only become feasible due to technological progress, which has gradually transformed human resource management from an operational process towards a strategic one (Karmańska 2020, 30; Zielinski 2023). While the classic day-to-day operations to meet employee needs used to be the main focus of HR, strategic future orientation in terms of developing human talent has gained tremendous importance (Chron 2020). In the same way, the data-driven approach of HR analytics results in HRM decisions being made less on a gut feeling and more based on statistically supported facts (van den Heuvel and Bondarouk 2017, 130; Brynjolfsson, Hitt and Heekyung 2011, 2). Analytics can help organizations optimize recruiting, budget HR costs, or measure employee performance more accurately (Lord 2018, 220; Karmańska 2020, 31). Sample use cases include the early detection of employees who are likely to churn, the optimized matching of skills and job profiles, or the pre-selection of applicants who best fit the company and the team in question (Hammermann, Lehr and Burstedde 2022, 5). In the area of compensation and benefits, the use of technology may, for example, enable the introduction of an automated reward system without the need for constant

supervision of employees, as relevant actions are recorded in data and credited automatically. (Ingale 2016).

However, the fundamental prerequisite for HR analytics to work is the availability of personnel-related data in digital form, as well as suitable methods for deriving decisions from this data (Hammermann, Lehr and Burstedde 2022, 5). Once HR data is available, data analysis can be performed in three different ways – descriptive, predictive, or prescriptive. Which of these methods is most suitable highly depends on the specific use case. Descriptive HR analytics, for instance, can help identify patterns by establishing a relationship between the current situation and historical data. Predictive analytics takes an additional step by inferring future trends from past developments up to the present. The most complex method is prescriptive analytics, which additionally forecasts possible consequences of the predicted data (Raghunadha Reddy and Lakshmikeerthi 2017, 26; Manikandan and Wilson 2023, 3093).

In conclusion, Lord (2018, 220) accurately underscores that the future's primary HRM focus will be on analytics, a priority that extends to various organizational types, including sports organizations, and the world of football.

3.4 Dashboard 3: Recruitment

Although the focus of this work is on compensation and benefits, other areas are closely connected to these concepts, including recruitment. As mentioned in the contextualization and literature review of this thesis, offering appropriate compensation and benefits is essential to attract qualified staff to the organization. For this reason, a dashboard showing recruiting KPIs was developed. The data used for this dashboard was obtained from the generated applicant dataset.

To start again with the filter options of the dashboard, it may be said that, compared to the two previous dashboards, one slicer has been replaced, namely the slicer to filter the organizational

form. With this, the data for the two divisions sports and corporate could be differentiated. However, for the recruitment data a greater emphasis was placed on the filtering of periods, so the previous slicer was replaced by one for the financial year. Using this filtering option, both time intervals and individual years can be viewed, which may either be determined using an integrated slider or entered manually. Moreover, an additional filter has been added below the dashboard title, which, when activated, only shows the data for the current financial year. It overrides the year slicer if a tick is set. In the case of the screenshot provided in the appendix, the tick is set for the current year, which means that all data only refers to 2023, even though the year slicer shows the period 2003 to 2023.

The left-hand side of the dashboard has been structured chronologically. The first visualization shows a classic pie chart, which provides information on how applicants became aware of a job posting by SLB. To allow a better comparison and classification of the number of applicants behind each slice of the pie, the labels were expressed as percentages. Underneath the pie chart, there is a funnel to be found. It maps the entire application process in a stepwise manner from application to hiring. The visual is called a funnel because data bars are stacked on top of each other from large to small, resulting in a funnel shape (Ferrari and Russo 2016, 271). It was considered most suitable for presenting the recruitment process because it accurately illustrates the loss of candidates within each step. At the bottom of the funnel, the conversion rate is displayed. In the case of the underlying artificial data, 41 of 509 applicants have been hired in the current year, which corresponds to a conversion of 8.1%.

Hence, the active pipeline below in turn only shows those candidates who have neither been hired nor rejected at the current point in time, i.e., all applicants who are still actively involved in the application process. A 100% stacked bar chart was used to achieve this visual representation, with the individual application steps being indicated by different colors. Assisted by the color codes in the legend, it becomes visible how many applicants are currently

in each step of the process.

At the top right of the dashboard, however, the focus is now back on currently active employees. A tree map is used here to show the composition of the current staff according to their contract type. As only four contract types were defined, the tree map in this case does not visually differ from the 100% stacked bar chart that was used for the active pipeline. However, these areas are not bars, but rectangles, which can also be stacked on top of each other horizontally depending on the data distribution. Tree maps are particularly suitable when many variables are involved, especially for illustrating hierarchies (Long et al. 2017, 109).

Before presenting the following illustrations, the two terms “time to hire” and “time to fill” first need to be distinguished from one another. Time to hire refers to the time that has passed between a candidate’s application and the signing of the contract. Days to fill, however, describe an even longer period, namely from the time of the job posting to the applicant’s first day of work (Xie 2020). First, an attempt was made to find a logic for calculating the “cost of vacancy”, i.e., the costs for the period in which a position is unfilled. To achieve this, the average annual compensation was divided by the average annual working days, reduced by absences. This is intended to represent the amount of money that an employee costs on average per working day. This amount is subsequently multiplied by the time to fill to determine what compensation the company would have been worth paying for an employee during the time to fill the position.

After that, the result is multiplied by a factor that is designed to reflect the importance of a position for overall business performance, i.e., the value an employee creates. In this case, random multipliers between 1 and 2.5 were selected depending on the advertised department and the type of contract. However, the importance of a position can be measured in all possible ways but is always hard to determine in a generalized monetary manner. When looking at the dashboard a factor of 1.0 can be noticed, which is because no filter was set for department and

contract type, on which the factor depends.

Finally, the average number of days to hire as well as the resulting average costs to hire for every age group are shown in two separate bar charts. The method used to calculate the cost to hire has already been explained based on data preparation in the methodology section above. To complete the dashboard, general average values for days to hire and cost to hire were shown below the column charts, irrespective of the age groups.

3.5 Dashboard 4: Performance

The next dashboard is about employee performance, in terms of hours worked, with a special focus on absences due to illness as well as overtime. Here the same four slicers as in the compensation and the benefits dashboard have been integrated. To emphasize the filter options of Power BI and the created dashboards, two filters were applied. For the gender slicer “female” was set and for the department “Scouting”, which shows that it is also possible to combine different filters. Furthermore, like the compensation dashboard, a tick can be placed here to display only active employees. However, as this extended filter option was not applied, the data of inactive employees was also reflected in the various graphs.

The first visual at the top left shows a clustered bar chart, which may seem a little misleading at first glance after the presentation of the previous dashboards. This is because the ranges shown on the y-axis do not represent the different age groups in this case, but rather categorize the number of days absent. The x-axis and thus the size of the bars now represents the number of employees who fall into the respective category. For example, the longest bar in this visual indicates that 20 employees were absent for a duration of between 16 and 20 days this year. To simplify the interpretation of this visual, the axis titles have been displayed as an exception this time.

The second graphic shows a matrix that matches the absence with the ratings from the employee

satisfaction survey. As indicated in the title of the visual, the values shown are the average rating scores. In other words, they show the average rating given for each combination of absence range and evaluation criterion. The cell at the bottom right for example can be interpreted as follows: employees who were absent for more than 25 days this year gave the work-life balance an average rating of 4.13. Again, 1 represents the worst possible rating score while 5 represents the best. To facilitate a rapid overview of the large number of values within this matrix, conditional formatting has been added as a visual aid. The coloration of the cells is gradual and depends on the rating scores, which means that cells with an average score of 4 or higher remain white, while those with a score of 1.5 or worse are displayed in full red. In this way, particularly poor ratings will stand out.

The bottom visualization on the left side of the dashboard depicts another scatter chart, this time comparing the annual days of absence with the annual benefit costs. In this visual, no data has been aggregated, i.e., each data point represents one employee. Likewise, it is not limited to the current year but includes all available data for female employees in scouting.

Now the graph on the top right is not only about employee absenteeism but also about overtime hours worked. The data is illustrated using a stacked column chart, with the number of overtime hours worked represented by the white columns and the absenteeism represented by the gray columns. For better comparability with the overtime hours, the absenteeism in this visual was also calculated on an hourly basis rather than daily. In addition, the data for absent hours has been negated in the background so that the columns grow downwards, while the columns for overtime grow upwards as normal. This results in a more vivid overall picture, which highlights the contrast between absent and additional hours more clearly. However, the difference between the two parameters is also shown underneath the columns by the black labels. Altogether, twelve columns are shown, each representing data for a different month in 2023 to make periodic decisions.

The closing figure on this dashboard shows another clustered column chart, but this time with several clusters for the first time. Among the five imaginary benefits defined for Sport Lisboa e Benfica are two that are related to employee's health, namely gym membership and health care. In this visual, the average hours of absence for three or rather 15 different groups of employees are shown. It is 15 because this visual again differentiates between age groups and there are three columns for each of the five age groups. The wine-red colored columns each represent the average hours of absence for employees who receive the gym membership benefit. The white columns in turn only include those employees who receive the benefit of health care. If an employee receives both the gym membership and health care benefits, their absence will also be included in both columns. Finally, the gray columns show the overall average hours of absence for all age groups, irrespective of the benefits received.

3.5.1 Dashboard 3: Recruitment

The next dashboard to be interpreted addresses the topic of recruitment. The overall purpose of this dashboard is to understand the staffing process and evaluate its effectiveness based on various time and cost factors.

To begin with, the various sources used by applicants to find vacancies at Sport Lisboa e Benfica were presented. The pie chart used shows briefly which source is most effective in attracting applicants. Especially in terms of cost savings, these metrics can be very valuable for a company. In the data example shown, more than a third of all applicants originated from LinkedIn, while only a mere 8% were referred by job centers. In such a case, the management could now allocate more budget to LinkedIn and less to job centers due to the great performance. However, the costs for the individual sources also need to be compared to assess how efficient such a budget reallocation would be.

The subsequent recruitment funnel provides a precise overview of the recruitment process and

can help the company identify any inconsistencies. An indication of inconsistencies could be the excessive loss of applicants at a certain stage of the process. For example, if it becomes apparent that only 5% of applicants are invited to work trials after completing an interview, this may be due to the recruiting staff being too selective. If such patterns can be detected, the cause should be investigated to prevent the unnecessary loss of applicants. The information shown in the recruitment funnel can also provide additional insight into the efficiency of individual applicant sources. When clicking on a slice of the pie chart with the sources, only data from applicants that can be assigned to this source will be considered for the entire dashboard. To get back to the example of LinkedIn and job centers, the conversion could also be compared here. It might be that more applicants reached SLB via LinkedIn, but ultimately more people were hired when they entered via job centers. In such a case, a more expensive source that attracts fewer applicants could still be more interesting if the quality of the applicants is demonstrably better and can increase conversion.

Underneath the funnel, the active pipeline adds information on how many applicants currently find themselves in which phase of the recruitment process. This provides the company with a certain degree of planning security when it comes to filling vacant positions. By using historical data from the recruitment funnel, it is possible to estimate how many positions will be filled by applicants in the pipeline. Here it can be particularly helpful to set a filter by department to see how much staff growth the company can expect in a particular area and where applicants may still need to be approached first.

The tree map at the top right of the dashboard additionally provides information on the composition of the current workforce in terms of their contract type. This overview can be used to easily monitor whether a certain level of diversity is maintained within the company. For example, part-time employees may be valuable as they could work more productively due to shorter working hours. Similarly, freelancers can be engaged in a very flexible and targeted

manner, while interns represent very cost-effective workers. This visual can therefore be an effective tool for SL Benfica to control this contract diversity within the company.

The following part of the dashboard mainly deals with the KPIs of time to hire, cost to hire, and cost of vacancy. How exactly the individual values were calculated has been explained in the dashboard description in section 4.3. The cost of vacancy is particularly interesting to demonstrate the approximate cost of an unfilled position to the managers. Even if this KPI is very difficult to accurately determine, it can be highly relevant when compared to the compensation expenses of an employee in that role. This comparison serves to underscore the criticality of filling a position. The time to hire and associated costs to hire now describe the average time and costs incurred per employee to fully complete the recruitment funnel. The two-column charts can then be used to highlight the complexity of the recruitment process for each age group. Taking the data generated by the dashboard as an example, it is easy to see that the youngest and oldest group on average went through the process more than 13 days faster than applicants between 50 and 66. As can also be observed from the cost to hire graph, this may result in lower costs, but it may also indicate that recruiters are making decisions too rashly. This data can help the management to intervene and therefore ensure the hiring of high-quality employees only.

3.5.2 Dashboard 4: Performance

The upcoming data to be discussed and interpreted is the data presented in the performance dashboard. The general purpose of this dashboard is to measure the performance of employees in various ways based on their hours worked, to gain potential insights into their motivation to work for SLB.

To begin with, the bar chart on overall absenteeism provides a clear overview of the distribution of sick days in the current year. This information can be particularly relevant for managers to

derive absence forecasts and thus improve resource planning in the company. If they know roughly how many employees will be absent for how long, they can react more effectively, for example by redistributing the workload or creating backup plans. Likewise, the visual can provide insights into the work climate and employee motivation. A high number of sick days might indicate potential problems within the company, such as excessive stress or a lack of employee satisfaction. Such cases can then be investigated in more detail to counteract these issues.

To be able to draw more reliable conclusions regarding the correlation between absenteeism and employee motivation, the ratings from the employee satisfaction survey have been included in the matrix below. By comparing the survey criteria individually with the respective absence ranges, it may be possible to identify trends in dissatisfaction to which these absences were related. The data shown in the dashboard does not provide a picture that would allow logical conclusions to be drawn. However, assuming that the work-life balance criterion was rated conspicuously poorly by employees with more than 25 days absence, for example at 1.5, this could be an indication that these employees feel overwhelmed and are therefore absent more often. As a consequence, SLB can intervene and take measures to counteract dissatisfaction and possibly reduce absenteeism. One approach could be to offer the benefit of flexible working hours to affected employees if they are not already benefiting from it.

The following scatter chart can be used to investigate a potential correlation between benefits received and employee absence. If a negative correlation can be identified here, i.e., the lower the value of the benefits, the higher the absenteeism, this may again indicate that benefits positively influence employee motivation. Of course, this is only one way to interpret these numbers. In addition to higher motivation, health-related benefits could also be the reason for lower absenteeism, for example. However, this aspect has been examined in more detail in the last visual of the dashboard.

Before continuing with this visual the added value of the stacked column chart on the upper right of the dashboard will be discussed. Here SL Benfica can get an impression of how many of the lost working hours could be compensated for by overtime worked each month. In the case of paid overtime, this also incurs costs for the company but therefore exerts a positive effect on productivity. This bar chart visualization is particularly useful in terms of resource planning. By providing data every month, seasonal trends such as increased sick leave in winter months might be identified. One corrective action to counter such a case would be to offer a bonus for overtime in these months to compensate for the lost hours with healthy employees.

As already indicated, the last visual of this dashboard examines the influence of health-related benefits on the absence rate of employees in the organization. If real data shows a picture like the fictional data in the dashboard screenshot, it can certainly be concluded that these benefits indeed have a positive effect on absenteeism. Across all age groups, the average absence rate is lower when employees have a gym membership and or health care access. Employees with a gym membership may have a stronger basic fitness level, while the health care benefit allows them to take actions such as preventive medical check-ups. If such a correlation can be observed, SL Benfica might consider offering these benefits to all employees as a way of reducing absenteeism. Of course, this should also be considered and evaluated from a cost perspective.

3.6 Recommendations and Potential Improvements

When considering recommendations for further research in the field of this work project, it is essential to first address the limitations previously described. Perhaps the major limitation in the course of this work – the absence of actual data – might potentially be resolved in the future through anonymization or aggregation. Instead of directly transmitting sensitive data, SLB could anonymize or aggregate data in a way that protects individual identities or confidential

details while retaining crucial patterns and trends.

Another approach might involve establishing contractual guidelines for data usage promptly. This way, third parties would be obliged to handle sensitive data appropriately and prevent its misuse. Measures such as these would enable the use of real-world data and thus allow a customized visualization strategy to be developed, which would greatly facilitate its subsequent implementation within the company. A potential access to actual data can therefore enable an assessment of the information already during the creation of the dashboards as well as a seamless implementation, as the underlying data structures do not have to be adapted to the company's formats first. Additionally, this scenario would provide insight into the current state of SL Benfica's KPI reporting, which would reduce the risk of distancing too far from the organization's existing visualization strategy through outside perspectives.

Irrespective of the authenticity of the data, future projects could consider excluding professional football players from the data and displaying their information in separate dashboards. In some cases, such as the compensation dashboard, the inclusion of these professionals can significantly distort the values, as they usually receive significantly higher salaries than corporate employees or the rest of the SLB sports staff. Even if professional players can be easily excluded using the built-in filters in the dashboards, this could lead to confusion under certain circumstances, which is why it would be worth considering splitting the data.

Thinking forward, it would be interesting to measure the emotional aspect of on-field performance, especially in the context of employee retention and loyalty or even recruitment. In other words, it might be worth investigating whether the success of the professional team on the pitch has a positive impact on these human resource management aspects. This impact may potentially be difficult to quantify and would therefore require extensive brainstorming on how to determine respective KPIs. However, if it was possible to identify a relationship of this kind, SLB could make use of it for its employer branding and thus gain a competitive advantage

(Urbancová and Hudáková 2017, 48).

The dashboards created in the course of this work aim to describe the personnel situation at Sport Lisboa e Benfica as precisely and clearly as possible to help the company in its decision-making. However, as already outlined in chapter 2.3, alongside this descriptive approach there are also predictive and prescriptive analytics that can be considered. As such, the visualization of HR metrics could be further optimized in future research including these approaches. One possible application could be the implementation of a machine learning model that predicts whether an employee will leave the organization in the upcoming financial year within a certain confidence interval (Malisetty and Kumari 2017, 117). Employees could be classified as either churners or non-churners based on independent input variables such as their tenure, educational level, and overall satisfaction score. Again, the creation of such a model involves a great deal of effort. To make the most accurate predictions possible, intensive research would be required to identify the most suitable and appropriate number of variables on which to build the model (Lamba and Madhusudhan 2022, 213).

Nevertheless, this can yield great added value. In the example of personnel loss, employees who are most likely to churn might be given extra incentives to stay such as a small salary increase. Furthermore, the quality of a predictive model generally improves over time as the amount of available data increases, which is why this is a recommendable extension for future projects of this kind.

4. References

Abdelbaki, Mohamed. 2022. *Effective data visualisation part 1: From numbers to insights.* September 24. Accessed November 08, 2023. https://new-narrative.com/2022/09/24/effective-data-visualisation-part-1-from-numbers-to-insights#:~:text=Edward%20Tufte%2C%20an%20American%20statistician,graphics%2C%20charts%2C%20and%20tables.

Armstrong, Michael. 2008. *Strategic Human Resource Management a Guide to Action.* London: Kogan Page Limited.

Aslam, Hassan Daniel, Muhammad Badar Habib, Naeem Ali, and Mehmood Aslam. 2014. "Importance of Human Resource Management in 21st Century: A Theoretical Perspective." *International Journal of Human Resource Studies* 3 (3): 87-96.

Bester, Vanessa. 2023. *indeed. What Are Compensation and Benefits? (A Complete Guide).* February 17. Accessed October 15, 2023. https://www.indeed.com/career-advice/pay-salary/compensation-and-benefits.

Bhanuprakash, Chandraiah, M. A. Jayaram, and Y. S. Nijagunarya. 2014. "A Simple Approach to SQL Joins in a Relational Algebraic Notation." *International Journal of Computer Applications* 104 (4): 975-8887.

Biswas, Bashker D. 2013. *Compensation and Benefit Design - Applying Finance and Accounting Principles to Global Human Resource Management Systems.* United States of America: Pearson Education.

Brynjolfsson, Erik, Lorin M. Hitt, and Hellen Kim Heekyung. 2011. "Strength in Numbers: How Does Data-Driven Decisionmaking Affect Firm Performance?" *SSRN Electronic Journal* 1-28.

Candradewi, Intan, and I Gst Ayu Manuati Dewi. 2019. "Effect of Compensation on Employee Performance towards Motivation as Mediation Variable." *International Research Journal of Management, IT & Social Sciences* 6 (5): 134-143.

Chron. 2020. *Operational HR Management Vs. Strategic HR Management.* July 29. Accessed October 18, 2023. https://smallbusiness.chron.com/operational-hr-management-vs-strategic-hr-management-62125.html.

Covell, Daniel, Sharianne Walker, Peter Hess, and Julie Siciliano. 2012. *Managing Sports Organizations.* Burlington: Butterworth-Heinemann.

Eessaar, Erki . 2016. "The Database Normalization Theory and the Theory of Normalized Systems: Finding a Common Ground." *Baltic Journal of Modern Computing* 4 (1): 5-33.

Enderes, Kathi. 2023. *LinkedIn. A Radically Different Approach to Pay and Benefits: Systemic Rewards.* May 05. Accessed October 16, 2023. https://www.linkedin.com/pulse/radically-different-approach-pay-benefits-systemic-rewards-enderes/.

Ferrari, Alberto, and Marco Russo. 2016. *Introducing Microsoft Power BI.* Redmond, Washington: Microsoft Press.

Gheorghe, Constantin. 2015. "Managers and Human Resource." *Revista Economică* 67 (5): 148-164.

Gholizadeh, Samira. 2022. "Top Popular Python Libraries in Research." *Journal of Robotics and Automation Research* 3 (2): 142-145.

Hammermann, Andrea, Judith Lehr, and Alexander Burstedde. 2022. *HR Analytics Anwendungsfelder und Erfolgsfaktoren.* IW-Report 28/2022, Köln: Institut der deutschen Wirtschaft Köln e. V.

Ingale, Yogesh. 2016. *LinkedIn. Moving from Operational HR Management to Strategic HR Management.* July 19. Accessed October 18, 2023. https://www.linkedin.com/pulse/moving-from-operational-hr-management-strategic-yogesh-ingale/.

Investing. 2023. *Financial Summary Benfica (SLBEN).* Accessed October 08, 2023. https://uk.investing.com/equities/sport-lisboa-benf-financial-summary.

Karmańska, Anna. 2020. "The benefits of HR analytics." *Prace Naukowe Uniwersytetu Ekonomicznego we Wrocławiu* 64 (8): 30-39.

Kraleva, Radoslava Stankova, Velin Kralev, Petia D Koprinkova-Hristova, Nadejda Bocheva, and Nina Sinyagina. 2018. "Design and Analysis of a Relational Database for Behavioral Experiments Data Processing." *International Journal of Online Engineering* 14 (2): 117-132.

Lamba, Manika, and Margam Madhusudhan. 2022. *Text Mining for Information Professionals: An Uncharted Territory.* Delhi: Springer Nature.

Long, Lim Kian , Lim Chien Hui, Gim Yeong Fook, Wan Mohd Nazmee, and Wan Zainon. 2017. "A Study on the Effectiveness of Tree-Maps as Tree Visualization Techniques." *Procedia Computer Science* 124: 108-115.

Lord, Jonathan. 2018. "Human resource management in football." In *Routledge Handbook of Football Business and Management*, by Simon Chadwick, Daniel Parnell, Paul Widdop and Christos Anagnostopoulos, 220-229. Abingdon: Routledge.

Malisetty, Sainath, and Vasanthi Kumari. 2017. "Predictive Analytics in HR Management." *Indian Journal of Public Health Research and Development* 8 (3): 116-120.

Manikandan, S., and Anitta Wilson. 2023. "HR Analytics: Importance and Benefits to Organizations." *International Journal of Research Publication and Reviews* 4 (7): 3091-3095.

Microsoft. n.d. *What is Power BI?* Accessed November 23, 2023. https://powerbi.microsoft.com/en-au/what-is-power-bi/.

Oladapo, Victor. 2014. "The Impact of Talent Management on Retention." *Journal of Business Studies Quarterly* 5 (3): 20-36.

Omar, Che Mohd Zulkifli Che. 2019. “Compensation and Benefit Key Reason Employees Retain.” *International Journal of Economics, Business and Management Research* 3 (4): 110-117.

PORDATA. n.d. Accessed November 25, 2023. https://www.pordata.pt/en/db/search+environment/new+search.

Price, Joey. 2023. *Forbes. Why HR Is Key To Executive Success: How The Human Resources Function Impacts Business Growth.* February 08. Accessed October 13, 2023. https://www.forbes.com/sites/forbeshumanresourcescouncil/2023/02/08/why-hr-is-key-to-executive-success-how-the-human-resources-function-impacts-business-growth/.

Raghunadha Reddy, P., and P. Lakshmikeerthi. 2017. “‘HR Analytics’ - An Effective Evidence Based HRM Tool.” *International Journal of Business and Management Invention* 6 (7): 23-34.

Reshma Ramdas, Nitnaware. 2019. “Basic Fundamental of Python Programming Language and The Bright Future.” *A Peer-Reviewed Journal About* 8 (2): 71-76.

Shtembari, Eriona, Andromahi Kufo, and Dea Haxhinasto. 2022. “Employee Compensation and Benefits Pre and Post COVID-19.” *Administrative Sciences* 12 (3): 1-17.

Urbancová, Hana, and Monika Hudáková. 2017. “Benefits of Employer Brand and the Supporting Trends.” *Economics and Sociology* 10 (4): 41-50.

van den Heuvel, Sjoerd, and Tanya Bondarouk. 2017. “The Rise (and Fall?) of HR Analytics: The Future Application, Value, Structure, and System Support.” *Journal of Organizational Effectiveness: People and Performance* 4 (2): 127-148.

van Vulpen, Erik. n.d. *AIHR. 7 Human Resource Management Basics Every HR Professional Should Know.* Accessed October 15, 2023. https://www.aihr.com/blog/human-resource-basics/.

W3Schools. n.d. *SQL FOREIGN KEY Constraint.* Accessed December 11, 2023. https://www.w3schools.com/sql/sql_foreignkey.asp.

Weerakoon, Ranjan Kumara. 2016. “Human Resource Management in Sports: A Critical Review of its Importance and Pertaining Issues.” *Physical Culture and Sport. Studies and Research* 69 (1): 15-21.

Xie, Jeslyn. 2020. *LinkedIn. Time-to-Fill vs. Time-to-Hire: Which Is More Important?* February 17. Accessed November 28, 2023. https://www.linkedin.com/pulse/time-to-fill-vs-time-to-hire-which-more-important-xie-chrp-cir-1c/.

Zielinski, Dave. 2023. *SHRM. The Evolution of HR and Technology.* June 02. Accessed October 18, 2023. https://www.shrm.org/hr-today/news/hr-magazine/summer-2023/pages/shrm-75th-technology.aspx.

5. Appendix

```python
import pandas as pd
import numpy as np
import random as rd
from datetime import datetime, timedelta
import warnings
warnings.filterwarnings("ignore")


### 1: EMPLOYEE MASTER DATA

employees = 20000
active_employees = 750
df = pd.DataFrame()

df["employee_id"] = df.index + 1
df["corporation"] = np.random.choice(["corporate","sport"], employees, p=[0.4, 0.6])
df["gender"] = np.random.choice(["male","female"], employees, p=[0.7, 0.3])
df["age"] = np.random.choice(["under 25","25-35","35-50","50-66","over 66"], employees, p=[0.18, 0.31, 0.27, 0.15, 0.09])

def age_start (age_str):
    if age_str == "over 66":
        return age_str[-2:]
    elif age_str == "under 25":
        return "18"
    else:
        return age_str[:2]

def age_end (age_str):
    if age_str == "over 66":
        return "80"
    else:
        return age_str[-2:]

df["age_interval_start"] = df["age"].apply(age_start).astype(int)
df["age_interval_end"] = df["age"].apply(age_end).astype(int)
df["educational_level"] = np.random.choice(["High School", "Bachelor's Degree", "Master's Degree", "PhD"], employees,
                                           p=[0.3, 0.4, 0.2, 0.1])
df["management_level"] = np.random.choice(["top", "middle", "first line", "support", "specialized technicians"],
                                          employees, p=[0.05, 0.1, 0.18, 0.61, 0.06])
df["type_of_contract"] = np.random.choice(["full time", "part time", "intern", "freelancer"], employees,
                                          p=[0.7, 0.2, 0.05, 0.05])
```

```
df["working_service"] = np.where((df["type_of_contract"] == "full time") |
                                 (df["type_of_contract"] == "part time") |
                                 (df["type_of_contract"] == "intern"),
                                 "fixed contract","occasional work")
df["department"] = np.where(df["corporation"] == "corporate", np.random.choice(["HR", "Marketing & Sales",
                            "Controlling", "Finance & Administration", "IT", "Media & Contents"], employees),
                            np.random.choice(["Scouting", "Youth Football", "Professional Football"], employees))

def generate_random_tenure(age_interval_end):
    upper_bound = (age_interval_end*12)-(18*12)
    return rd.randint(1, upper_bound)

df['tenure_in_months'] = df['age_interval_end'].apply(generate_random_tenure)
df["tenure_in_years"] = (df["tenure_in_months"] / 12).astype(int)

df['active_employee'] = 0
active_indices = rd.sample(range(len(df)), active_employees)
df.loc[active_indices, "active_employee"] = 1

df['terminated_by_employee'] = np.where(df['active_employee'] == 0, np.random.choice([1, 0], employees, p=[0.85, 0.15]), None)
df["termination_reason"] = np.where(df["terminated_by_employee"] == 1,
                                    np.random.choice(["New Professional Challenge", "Salary Package",
                                                      "Lack of Growth Perspective", "Work Schedule",
                                                      "Family Reasons", "Contractual Bond", "Workplace",
                                                      "Conflict with Management", "Conflict with Colleagues",
                                                      "Very Procedural and Repetitive Functions", "Personal Reasons",
                                                      "Expectations not Met", "Search for Professional Development",
                                                      "Return to Studies"], employees,
                                                      p=[0.23, 0.17, 0.15, 0.05, 0.08, 0.02, 0.03, 0.01, 0.02, 0.01,
                                                         0.16, 0.02, 0.04, 0.01]), None)

def calculate_start_date(row):
    tenure_in_months = row['tenure_in_months']
    active_employee = row['active_employee']
    if active_employee == 1:
        start_date = datetime.now() - timedelta(days=tenure_in_months * 30)  # Approximating 30 days per month
    else:
        start_date_min = datetime(1963, 1, 1)
        start_date_max = datetime.now() - timedelta(days=tenure_in_months * 30)
        if start_date_min < start_date_max:
            start_date = start_date_min + timedelta(days=rd.randint(0, (start_date_max - start_date_min).days))
        else:
            start_date = start_date_min
    return start_date.date()
```

```
df['start'] = df.apply(calculate_start_date, axis=1)
df["end"] = df.apply(lambda row: pd.to_datetime(row["start"]) + timedelta(days=row["tenure_in_years"] * 365)
                          if row["active_employee"] == 0 else None, axis=1)

df["start"] = df["start"].apply(lambda x: x.year)
df["end"] = df["end"].apply(lambda x: x.year)
df['end'] = df['end'].fillna(0).astype(int)

df["benefit_healthcare"] = np.random.choice([1, 0], employees)
df["benefit_flexible_work"] = np.random.choice([1, 0], employees)
df["benefit_gym_membership"] = np.random.choice([1, 0], employees)
df["benefit_navegante"] = np.random.choice([1, 0], employees)
df["benefit_benfica_membership"] = np.random.choice([1, 0], employees)

def score(row):
    if row['end'] == 2023:
        return np.random.choice([1, 2, 3, 4, 5], p=[0.4, 0.35, 0.1, 0.1, 0.05])
    else:
        return np.random.choice([1, 2, 3, 4, 5], p=[0.05, 0.1, 0.25, 0.3, 0.3])

df["score_work_life_balance"] = df.apply(score, axis=1)
df["score_colleagues"] = df.apply(score, axis=1)
df["score_compensation"] = df.apply(score, axis=1)
df["score_career_opportunities"] = df.apply(score, axis=1)
df["score_job_tasks"] = df.apply(score, axis=1)

df["employee_id"] = df.index + 1


### 2: HISTORICAL EMPLOYEE DATA

df_hist = df
duplicated_rows = []

for _, row in df_hist.iterrows():
        duplicated_rows.extend([row] * (row["tenure_in_years"] + 1))

df_hist = pd.DataFrame(duplicated_rows, columns=df_hist.columns).reset_index(drop=True)

df_hist["year"] = df_hist["start"]

def increment_years(group):
    group["year"] += np.arange(len(group))
```

```
    return group

df_hist = df_hist.groupby("employee_id").apply(increment_years)
df_hist = df_hist.reset_index(drop=True)

df_hist = df_hist[df_hist.year >= 2005]

df_hist = df_hist.loc[df_hist.index.repeat(12)].reset_index(drop=True)
df_hist.reset_index(drop=True, inplace=True)

df_hist['month_number'] = df_hist.groupby(['employee_id', 'year']).cumcount() % 12 + 1

def mon(row):
    if row["month_number"] == 1:
        return "Jan"
    if row["month_number"] == 2:
        return "Feb"
    if row["month_number"] == 3:
        return "Mar"
    if row["month_number"] == 4:
        return "Apr"
    if row["month_number"] == 5:
        return "May"
    if row["month_number"] == 6:
        return "Jun"
    if row["month_number"] == 7:
        return "Jul"
    if row["month_number"] == 8:
        return "Aug"
    if row["month_number"] == 9:
        return "Sep"
    if row["month_number"] == 10:
        return "Oct"
    if row["month_number"] == 11:
        return "Nov"
    if row["month_number"] == 12:
        return "Dec"

df_hist["month"] = df_hist.apply(mon, axis=1)

inactive_rows = df_hist[df_hist['active_employee'] == 0].copy()

inactive_rows.sort_values(['employee_id', 'year', 'month_number'], ascending=[True, False, False], inplace=True)
```

```
inactive_rows['drop'] = 0  # Initialize the drop column
for _, group in inactive_rows.groupby('employee_id'):
    num_ones = rd.randint(0, 11)
    if num_ones > 0:
        last_rows = group.head(num_ones).index
        inactive_rows.loc[last_rows, 'drop'] = 1

df_hist['drop'] = np.where(df_hist['active_employee'] == 1, 0, inactive_rows['drop'].reindex(df_hist.index, fill_value=0))

df_hist = df_hist[df_hist["drop"] == 0]

inactive_rows = df_hist[df_hist['start'] == df_hist['year']].copy()
inactive_rows.sort_values(['employee_id', 'year', 'month_number'], ascending=[True, True, True], inplace=True)
inactive_rows['drop'] = 0  # Initialize the drop column

for _, group in inactive_rows.groupby('employee_id'):
    num_ones = rd.randint(0, 11)
    if num_ones > 0:
        last_rows = group.head(num_ones).index
        inactive_rows.loc[last_rows, 'drop'] = 1

df_hist['drop'] = np.where(df_hist['active_employee'] == 1, 0, inactive_rows['drop'].reindex(df_hist.index, fill_value=0))

df_hist = df_hist[df_hist["drop"] == 0]
df_hist = df_hist.drop(columns="drop")

salary_dict = {}

for index, row in df_hist.iterrows():
    employee_id = row["employee_id"]
    level = row["management_level"]
    if level == "top":
        salary_value = rd.randint(4200, 6000)
    elif level == "middle":
        salary_value = rd.randint(2900, 4200)
    elif level == "first line":
        salary_value = rd.randint(2300, 3300)
    else:
        salary_value = rd.randint(1600, 2300)
    salary_dict[employee_id] = salary_value

df_hist["base_salary"] = df_hist["employee_id"].map(salary_dict)

def benefits(row):
```

```
    x = 0
    if row["benefit_healthcare"] == 1:
        x += 120
    if row["benefit_flexible_work"] == 1:
        x += 50
    if row["benefit_gym_membership"] == 1:
        x += 35
    if row["benefit_navegante"] == 1:
        x += 40
    if row["benefit_benfica_membership"] == 1:
        x += 35
    return x

df_hist["monthly_benefits"] = df_hist.apply(benefits, axis=1)
df_hist["salary_plus_benefits"] = df_hist["base_salary"] + df_hist["monthly_benefits"]

df_hist["promotion"] = 0
promotions_counter = {}

for index, row in df_hist.iterrows():
    employee_id = row["employee_id"]
    if promotions_counter.get(employee_id, 0) < (row["tenure_in_years"] / 4):
        promotion_value = np.random.choice([0, 1], p=[0.7, 0.3])
        df_hist.at[index, 'promotion'] = promotion_value
        promotions_counter[employee_id] = promotions_counter.get(employee_id, 0) + promotion_value

df_hist['promotion'] = df_hist['promotion'].astype(int)

def calculate_salary_increase(row):
    if row['promotion'] == 1:
        increase_percentage = np.random.uniform(0.01, 0.10)
        return row['base_salary'] * increase_percentage
    else:
        return 0

df_hist['salary_increase'] = df_hist.apply(calculate_salary_increase, axis=1)
df_hist['salary_increase'] = df_hist['salary_increase'].astype(int)
df_hist["salary_increase_rel"] = round(df_hist["salary_increase"] / df_hist["salary_plus_benefits"], 2)
df_hist['full_salary'] = df_hist.groupby('employee_id')['salary_increase'].cumsum() + df_hist['salary_plus_benefits']

def reg_hours(emp):
    if (emp == "full time") | (emp == "intern"):
        return 4*40
    if emp == "part time":
```

```
        return 4*20
    if emp == "freelancer":
        return rd.randrange(8, 4*40, 8)

df_hist["regular_hours"] = df_hist["type_of_contract"].apply(reg_hours)

df_hist["vacation_hours"] = (df_hist["regular_hours"] * (30*8)/(12*4*40)).astype(int)

def sick_hours(row):
    max_hours = int((row["regular_hours"] - row["vacation_hours"])*0.3)
    return rd.randint(0, max_hours)

df_hist["sickness_hours"] = df_hist.apply(sick_hours, axis=1)
df_hist["absenteeism_in_days"] = (df_hist["vacation_hours"] + df_hist["sickness_hours"]) / 8

def over(reg):
    return rd.randint(0, int(reg*0.18))

df_hist["overtime_hours"] = df_hist["regular_hours"].apply(over)

df_hist = df_hist.drop(columns=["corporation", "gender", "age", "age_interval_start",
                                "age_interval_end", "educational_level", "type_of_contract",
                                "working_service", "department", "management_level",
                                "tenure_in_months", "tenure_in_years", "terminated_by_employee",
                                "termination_reason", "benefit_healthcare", "benefit_flexible_work",
                                "benefit_gym_membership", "benefit_navegante", "benefit_benfica_membership"])

df_hist = df_hist[df_hist.year <= 2023]
df_hist = df_hist[df_hist.end <= 2023]

df_hist = df_hist.sort_values(by=['employee_id', 'year', 'month_number'])
df_hist['starter'] = df_hist.groupby('employee_id').cumcount() == 0
df_hist['starter'] = df_hist['starter'].astype(int)
df_hist.reset_index(drop=True, inplace=True)

df_hist = df_hist.sort_values(by=['employee_id', 'year', 'month_number'])
df_hist['leaver'] = ((df_hist.groupby('employee_id')['active_employee'].transform('last') == 0)
                     & (df_hist.groupby('employee_id').cumcount(ascending=False) == 0))
df_hist["leaver"] = df_hist['leaver'].astype(int)
df_hist.reset_index(drop=True, inplace=True)


### 3: APPLICANT DATA
```

```
applicants = 50000
hired = 20000

employee_ids = np.random.permutation(range(1, hired + 1))
df_app = pd.DataFrame({'employee_id': employee_ids[:hired]})
df_app = df_app.append(pd.DataFrame({'employee_id': [0] * (applicants - hired)}), ignore_index=True)

df_app = pd.merge(df_app, df[["employee_id", "start", "gender", "age", "department", "corporation", "educational_level",
                              "type_of_contract", "management_level"]], on='employee_id', how='left')

def start(row):
    if row["employee_id"] == 0:
        return rd.randint(1963,2023)
    else:
        return int(row["start"])

df_app["start"] = df_app.apply(start, axis=1)

def gender(row):
    if row["employee_id"] == 0:
        return np.random.choice(["male", "female"], p=[0.7, 0.3])
    else:
        return row["gender"]

df_app["gender"] = df_app.apply(gender, axis=1)

def age(row):
    if row["employee_id"] == 0:
        return np.random.choice(["under 25", "25-35", "35-50", "50-66", "over 66"])
    else:
        return row["age"]

df_app["age"] = df_app.apply(age, axis=1)

def corp(row):
    if row["employee_id"] == 0:
        return np.random.choice(["corporate", "sport"], p=[0.4, 0.6])
    else:
        return row["corporation"]

df_app["corporation"] = df_app.apply(corp, axis=1)

def dep(row):
    if (row["employee_id"] == 0) & (row["corporation"] == "corporate"):
```

```
        return np.random.choice(["HR", "Marketing & Sales", "Controlling", "Finance & Administration", "IT",
                                 "Media & Contents"])
    elif (row["employee_id"] == 0) & (row["corporation"] == "sport"):
        return np.random.choice(["Scouting", "Youth Football", "Professional Football"])
    else:
        return row["department"]

df_app["department"] = df_app.apply(dep, axis=1)

def edu(row):
    if row["employee_id"] == 0:
        return np.random.choice(["High School", "Bachelor's Degree", "Master's Degree", "PhD"], p=[0.3, 0.4, 0.2, 0.1])
    else:
        return row["educational_level"]

df_app["educational_level"] = df_app.apply(edu, axis=1)

def con(row):
    if row["employee_id"] == 0:
        return np.random.choice(["full time", "part time", "intern", "freelancer"], p=[0.7, 0.2, 0.05, 0.05])
    else:
        return row["type_of_contract"]

df_app["type_of_contract"] = df_app.apply(con, axis=1)

def man(row):
    if row["employee_id"] == 0:
        return np.random.choice(["top", "middle", "first line", "support", "specialized technicians"],
                                p=[0.02, 0.64, 0.1, 0.18, 0.06])
    else:
        return row["management_level"]

df_app["management_level"] = df_app.apply(man, axis=1)

df_app["source"] = np.random.choice(["LinkedIn", "SLB Employee", "SLB Website", "Job Center", "Friends & Family", "Other"],
                                    applicants, p=[0.35,0.04,0.19,0.08,0.21,0.13])

def step5(row):
    if row["employee_id"] != 0:
        return 1
    else:
        return 0

df_app["hired"] = df_app.apply(step5, axis=1)
```

```
def step4(row):
    if row["hired"] == 1:
        return 1
    else:
        return rd.randint(0,1)

df_app["trial_work"] = df_app.apply(step4, axis=1)

def step3(row):
    if row["trial_work"] == 1:
        return 1
    else:
        return rd.randint(0,1)

df_app["interview"] = df_app.apply(step3, axis=1)

def step2(row):
    if row["interview"] == 1:
        return 1
    else:
        return rd.randint(0,1)

df_app["screening"] = df_app.apply(step2, axis=1)

def offer(row):
    if (row["trial_work"] == 1) & (row["hired"] == 0):
        return np.random.choice([0,1], p=[0.73,0.27])
    else:
        return 0

df_app["offer_rejected"] = df_app.apply(offer, axis=1)

df_app["time_hired"] = df_app.apply(lambda x: np.random.randint(1, 17) if x["hired"] == 1 else 0, axis=1)
df_app["time_trial_work"] = df_app.apply(lambda x: np.random.randint(1, 19) if x["trial_work"] == 1 else 0, axis=1)
df_app["time_interview"] = df_app.apply(lambda x: np.random.randint(1, 20) if x["interview"] == 1 else 0, axis=1)
df_app["time_screening"] = df_app.apply(lambda x: np.random.randint(1, 26) if x["screening"] == 1 else 0, axis=1)
df_app["time_application"] = df_app.apply(lambda x: np.random.randint(7, 26), axis=1)

df_app["time_to_hire"] = df_app["time_screening"]+df_app["time_interview"]+df_app["time_trial_work"]+df_app["time_hired"]
df_app["time_to_fill"] = df_app["time_to_hire"]+df_app["time_application"]


df_app["cost_hired"] = df_app.apply(lambda x: 1200+x["time_hired"]*10 if x["hired"] == 1 else 0, axis=1)
```

```
df_app["cost_trial_work"] = df_app.apply(lambda x: 230+x["time_trial_work"]*10 if x["trial_work"] == 1 else 0, axis=1)
df_app["cost_interview"] = df_app.apply(lambda x: 160+x["time_interview"]*10 if x["interview"] == 1 else 0, axis=1)
df_app["cost_screening"] = df_app.apply(lambda x: 70+x["time_screening"]*10 if x["screening"] == 1 else 0, axis=1)

df_app["cost_to_hire"] = df_app["cost_screening"]+df_app["cost_interview"]+df_app["cost_trial_work"]+df_app["cost_hired"]

df.to_csv("benfica_data.csv")
df_hist.to_csv("emp_history.csv")
df_app.to_csv("emp_app.csv")

df
```

	employee_id	corporation	gender	age	age_interval_start	age_interval_end	educational_level	management_level	type_of_contract	working_se
0	1	sport	male	35-50	35	50	Bachelor's Degree	support	full time	fixed cor
1	2	corporate	female	over 66	66	80	Master's Degree	support	full time	fixed cor
2	3	corporate	male	50-66	50	66	PhD	support	full time	fixed cor
3	4	corporate	male	25-35	25	35	High School	support	part time	fixed cor
4	5	sport	male	under 25	18	25	Bachelor's Degree	top	full time	fixed cor
...	...	...	...	...	...	...	...	...	...	
19997	19998	sport	female	under 25	18	25	Master's Degree	middle	part time	fixed cor
19998	19999	sport	female	35-50	35	50	PhD	support	full time	fixed cor
19999	20000	sport	male	50-66	50	66	Master's Degree	middle	part time	fixed cor

20000 rows × 28 columns

df_hist

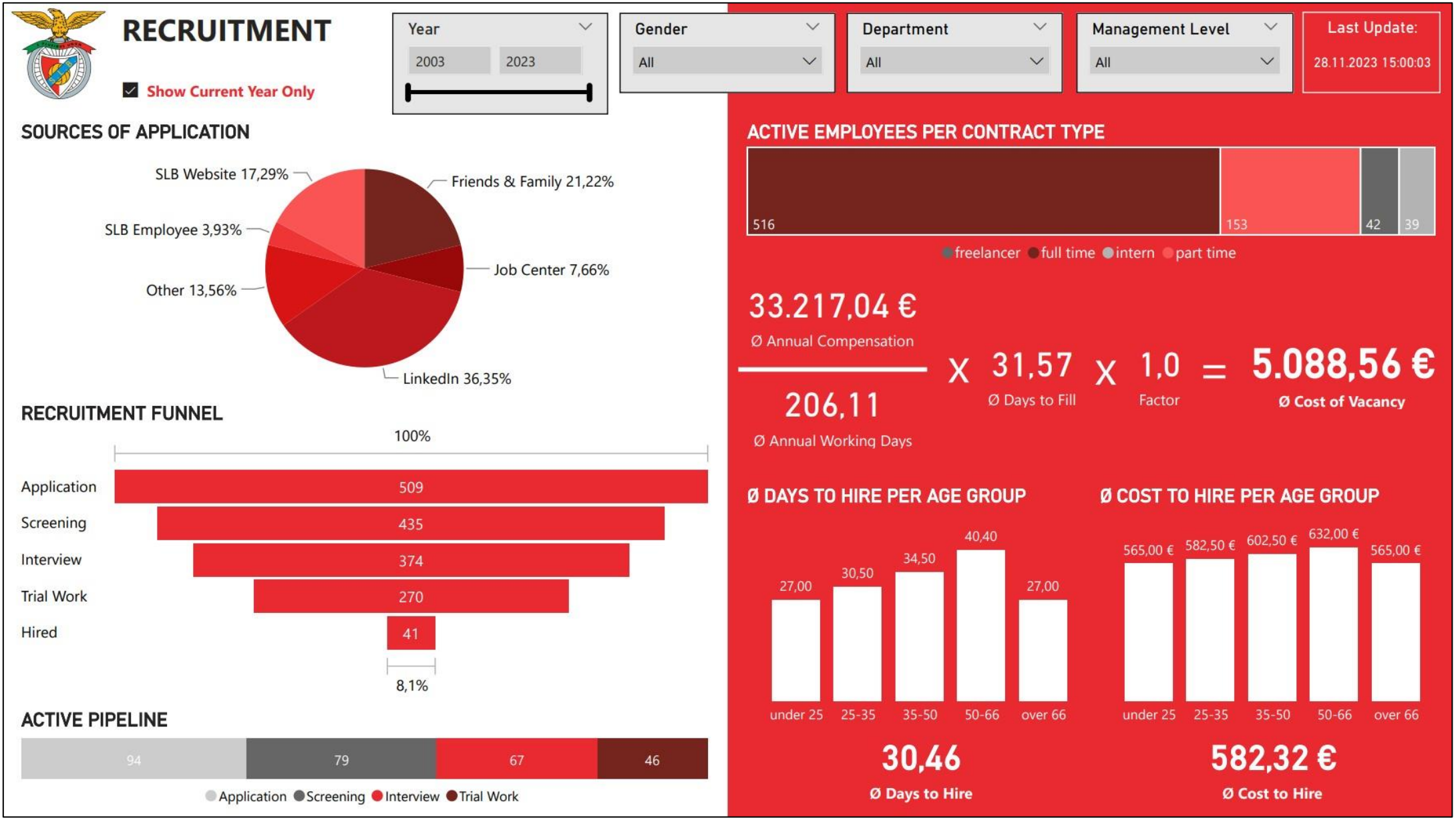
	employee_id	active_employee	start	end	year	month_number	month	base_salary	annual_benefits	salary_plus_benefits	...	salary_increase
0	1	0	1963	2023	2005	1	Jan	2070	90	2160	...	36
1	1	0	1963	2023	2005	2	Feb	2070	90	2160	...	0
2	1	0	1963	2023	2005	3	Mar	2070	90	2160	...	0
3	1	0	1963	2023	2005	4	Apr	2070	90	2160	...	0
4	1	0	1963	2023	2005	5	May	2070	90	2160	...	0
...	...	...	...	...	...	...	...	...	...	...	...	...
835470	19998	0	2020	2020	2020	7	Jul	2069	280	2349	...	0

835471 rows × 21 columns

df_app

	employee_id	start	gender	age	department	corporation	educational_level	type_of_contract	management_level	source	...	time_interviev
0	1321	1991	male	35-50	Finance & Administration	corporate	High School	full time	support	Job Center	...	1
1	6051	1999	male	25-35	Scouting	sport	High School	full time	first line	LinkedIn	...	1
2	17035	1973	female	50-66	Scouting	sport	Master's Degree	part time	support	Friends & Family	...	
3	3277	1974	male	35-50	Finance & Administration	corporate	High School	part time	first line	Friends & Family	...	1
4	938	1963	female	25-35	IT	corporate	Bachelor's Degree	full time	first line	LinkedIn	...	1
...	...	...	...	...	...	...	...	...	...	...	...	
49999	0	2004	male	under 25	Scouting	sport	Bachelor's Degree	freelancer	support	LinkedIn	...	

50000 rows × 27 columns

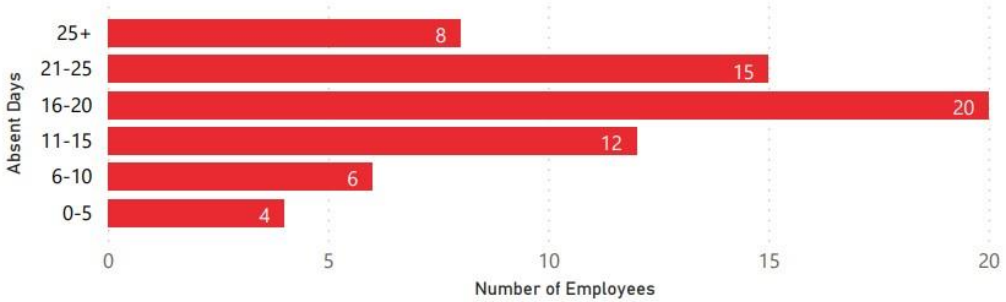

RECRUITMENT
Show Current Year Only
Year
2003
2023
Gender
All
Department
All
Management Level
All
Last Update:
28.11.2023 15:00:03
SOURCES OF APPLICATION
SLB Website 17,29%
Friends & Family 21,22%
SLB Employee 3,93%
Job Center 7,66%
Other 13,56%
LinkedIn 36,35%
RECRUITMENT FUNNEL
100%
Application 509
Screening 435
Interview 374
Trial Work 270
Hired 41
8,1%
ACTIVE PIPELINE
94
79
67
46
Application Screening Interview Trial Work
ACTIVE EMPLOYEES PER CONTRACT TYPE
516
153
42
39
freelancer full time intern part time
33.217,04 €
Ø Annual Compensation
206,11
Ø Annual Working Days
X 31,57
Ø Days to Fill
X 1,0
Factor
= 5.088,56 €
Ø Cost of Vacancy
Ø DAYS TO HIRE PER AGE GROUP
27,00
30,50
34,50
40,40
27,00
under 25
25-35
35-50
50-66
over 66
30,46
Ø Days to Hire
Ø COST TO HIRE PER AGE GROUP
565,00 €
582,50 €
602,50 €
632,00 €
565,00 €
under 25
25-35
35-50
50-66
over 66
582,32 €
Ø Cost to Hire

PERFORMANCE

Gender: female | Department: Scouting | Organization: All | Management Level: All | Last Update: 29.11.2023 01:09:56

☐ Show Active Employees Only

DAYS OF ABSENTEEISM 2023

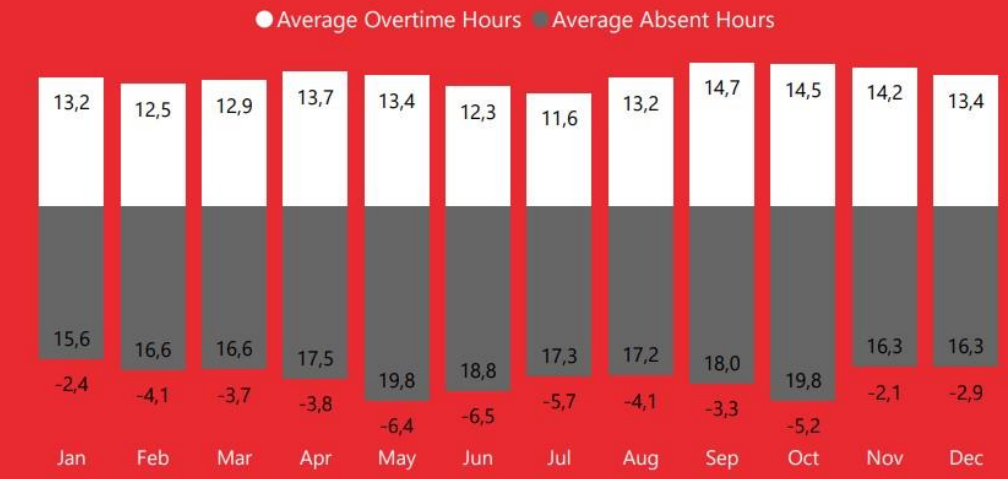


Ø RATINGS VS ABSENTEEISM

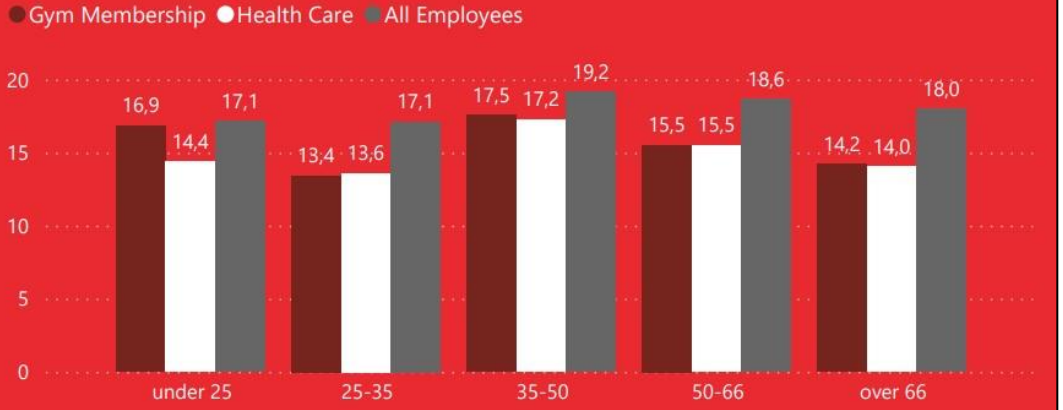
Absent Days	Career Opportunities	Colleagues	Compensation	Job Tasks	Work Life Balance
0-5	1,50	1,75	2,25	1,75	3,50
6-10	1,17	2,50	2,67	2,00	1,67
11-15	3,08	2,75	2,67	2,83	3,33
16-20	3,00	3,10	3,25	3,05	3,35
21-25	3,53	4,07	3,67	3,27	2,93
25+	3,38	3,63	3,75	3,25	4,13

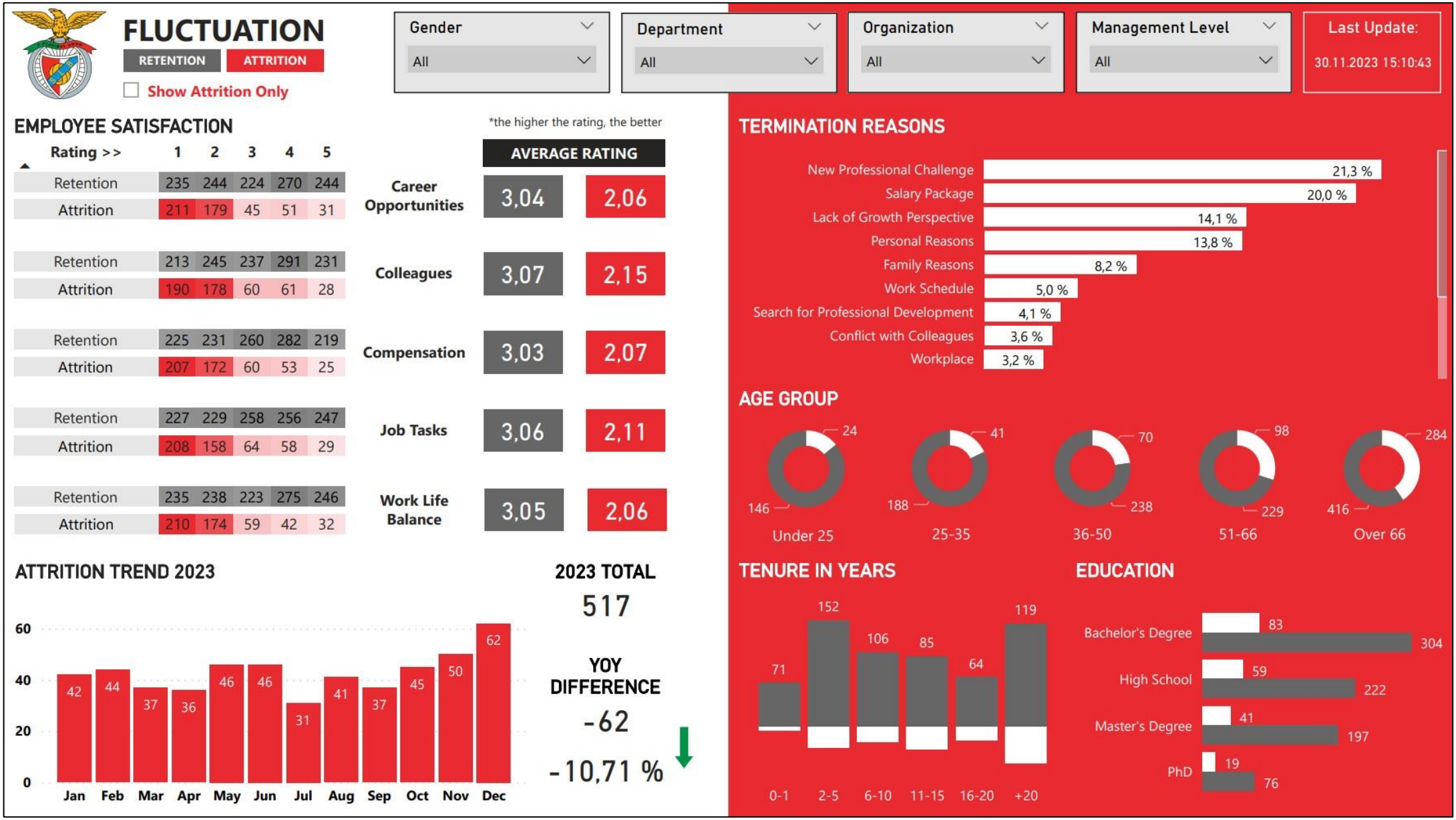
ANNUAL ABSENT DAYS VS ANNUAL BENEFIT COSTS

Ø OVERTIME HOURS VS Ø ABSENT HOURS 2023



Ø ABSENT HOURS WITH HEALTH RELATED BENEFITS PER AGE GROUP




FLUCTUATION
RETENTION
ATTRITION
Show Attrition Only
Gender
All
Department
All
Organization
All
Management Level
All
Last Update:
30.11.2023 15:10:43
EMPLOYEE SATISFACTION
*the higher the rating, the better
Rating >> 1 2 3 4 5
AVERAGE RATING
Retention 235 244 224 270 244
Attrition 211 179 45 51 31
Career Opportunities
3,04
2,06
Retention 213 245 237 291 231
Attrition 190 178 60 61 28
Colleagues
3,07
2,15
Retention 225 231 260 282 219
Attrition 207 172 60 53 25
Compensation
3,03
2,07
Retention 227 229 258 256 247
Attrition 208 158 64 58 29
Job Tasks
3,06
2,11
Retention 235 238 223 275 246
Attrition 210 174 59 42 32
Work Life Balance
3,05
2,06
TERMINATION REASONS
New Professional Challenge 21,3 %
Salary Package 20,0 %
Lack of Growth Perspective 14,1 %
Personal Reasons 13,8 %
Family Reasons 8,2 %
Work Schedule 5,0 %
Search for Professional Development 4,1 %
Conflict with Colleagues 3,6 %
Workplace 3,2 %
AGE GROUP
24 146 Under 25
41 188 25-35
70 238 36-50
98 229 51-66
284 416 Over 66
ATTRITION TREND 2023
60
40
20
0
Jan 42
Feb 44
Mar 37
Apr 36
May 46
Jun 46
Jul 31
Aug 41
Sep 37
Oct 45
Nov 50
Dec 62
2023 TOTAL
517
YOY DIFFERENCE
-62
-10,71 %
TENURE IN YEARS
71 0-1
152 2-5
106 6-10
85 11-15
64 16-20
119 +20
EDUCATION
Bachelor's Degree 83 304
High School 59 222
Master's Degree 41 197
PhD 19 76