Exploratory Data Analysis - Diabetes Dataset

In [1]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns

from med_project.config import RAW_DATA_DIR
import med_project.plots as plots
```

In [2]:
```
diabetes = pd.read_csv(RAW_DATA_DIR / 'diabetes_dataset.csv').drop(
    ['diabetes_stage'], axis=1
)
```

We're loading the dataset and dropping the 'diabetes_stage' column. This categorical variable tells us what stage of diabetes the patient is in (No Diabetes, Prediabetes, Type 1, Type 2).

Our goal is to build a regression model that predicts 'diabetes_risk_score', which is a continuous measure of diabetes risk. Using 'diabetes_stage' as a feature would be data leakage since it's essentially another representation of the outcome we're trying to predict. The diabetes stage might be determined based on the same health indicators that contribute to the risk score, so including it would give our model unfair information it wouldn't have in a real prediction scenario.

Basic Data Overview

In [3]:
```
diabetes.head()
```

Out[3]:

	age	gender	ethnicity	education_level	income_level	employment_status	smoking
0	58	Male	Asian	Highschool	Lower-Middle	Employed	
1	48	Female	White	Highschool	Middle	Employed	
2	60	Male	Hispanic	Highschool	Middle	Unemployed	
3	74	Female	Black	Highschool	Low	Retired	
4	46	Male	White	Graduate	Middle	Retired	

5 rows × 30 columns

Looking at the first few rows, we can see a good variety in the data - different ages, genders, ethnicities, and health profiles. The data types look appropriate with categorical variables stored as objects and numerical measurements as numeric types. We have both our target variable 'diabetes_risk_score' and the binary 'diagnosed_diabetes' column still present at this stage.

Data types

```
In [4]: diabetes.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 30 columns):
 #   Column                              Non-Null Count   Dtype  
---  ------                              --------------   -----  
 0   age                                 100000 non-null  int64  
 1   gender                              100000 non-null  object 
 2   ethnicity                           100000 non-null  object 
 3   education_level                     100000 non-null  object 
 4   income_level                        100000 non-null  object 
 5   employment_status                   100000 non-null  object 
 6   smoking_status                      100000 non-null  object 
 7   alcohol_consumption_per_week        100000 non-null  int64  
 8   physical_activity_minutes_per_week  100000 non-null  int64  
 9   diet_score                          100000 non-null  float64
 10  sleep_hours_per_day                 100000 non-null  float64
 11  screen_time_hours_per_day           100000 non-null  float64
 12  family_history_diabetes             100000 non-null  int64  
 13  hypertension_history                100000 non-null  int64  
 14  cardiovascular_history              100000 non-null  int64  
 15  bmi                                 100000 non-null  float64
 16  waist_to_hip_ratio                  100000 non-null  float64
 17  systolic_bp                         100000 non-null  int64  
 18  diastolic_bp                        100000 non-null  int64  
 19  heart_rate                          100000 non-null  int64  
 20  cholesterol_total                   100000 non-null  int64  
 21  hdl_cholesterol                     100000 non-null  int64  
 22  ldl_cholesterol                     100000 non-null  int64  
 23  triglycerides                       100000 non-null  int64  
 24  glucose_fasting                     100000 non-null  int64  
 25  glucose_postprandial                100000 non-null  int64  
 26  insulin_level                       100000 non-null  float64
 27  hba1c                               100000 non-null  float64
 28  diabetes_risk_score                 100000 non-null  float64
 29  diagnosed_diabetes                  100000 non-null  int64  
dtypes: float64(8), int64(16), object(6)
memory usage: 22.9+ MB
```

The dataset contains 100,000 records with 30 features (after dropping diabetes_stage). We have categorical variables like gender, ethnicity, education, and employment stored as objects. Numerical measurements like age, BMI, blood pressure, and lab values are stored as integers or floats. Binary variables like family history and medical conditions are stored as integers (0/1). This is a good starting point, though we'll need to encode the categorical variables properly before modeling.

Missing and duplicated data

```
In [5]: print(f'Missing data rows: {diabetes.isna().sum().sum()}')
```

```
Missing data rows: 0
```

```
In [6]: print(f'Duplicated values count: {diabetes.duplicated().sum()}')
```

```
Duplicated values count: 0
```

The dataset has no missing values or duplicate records, so we can proceed directly with the analysis.

Numerical data description

```
In [7]: diabetes.describe().drop('count', axis=0).T
```

Out[7]:

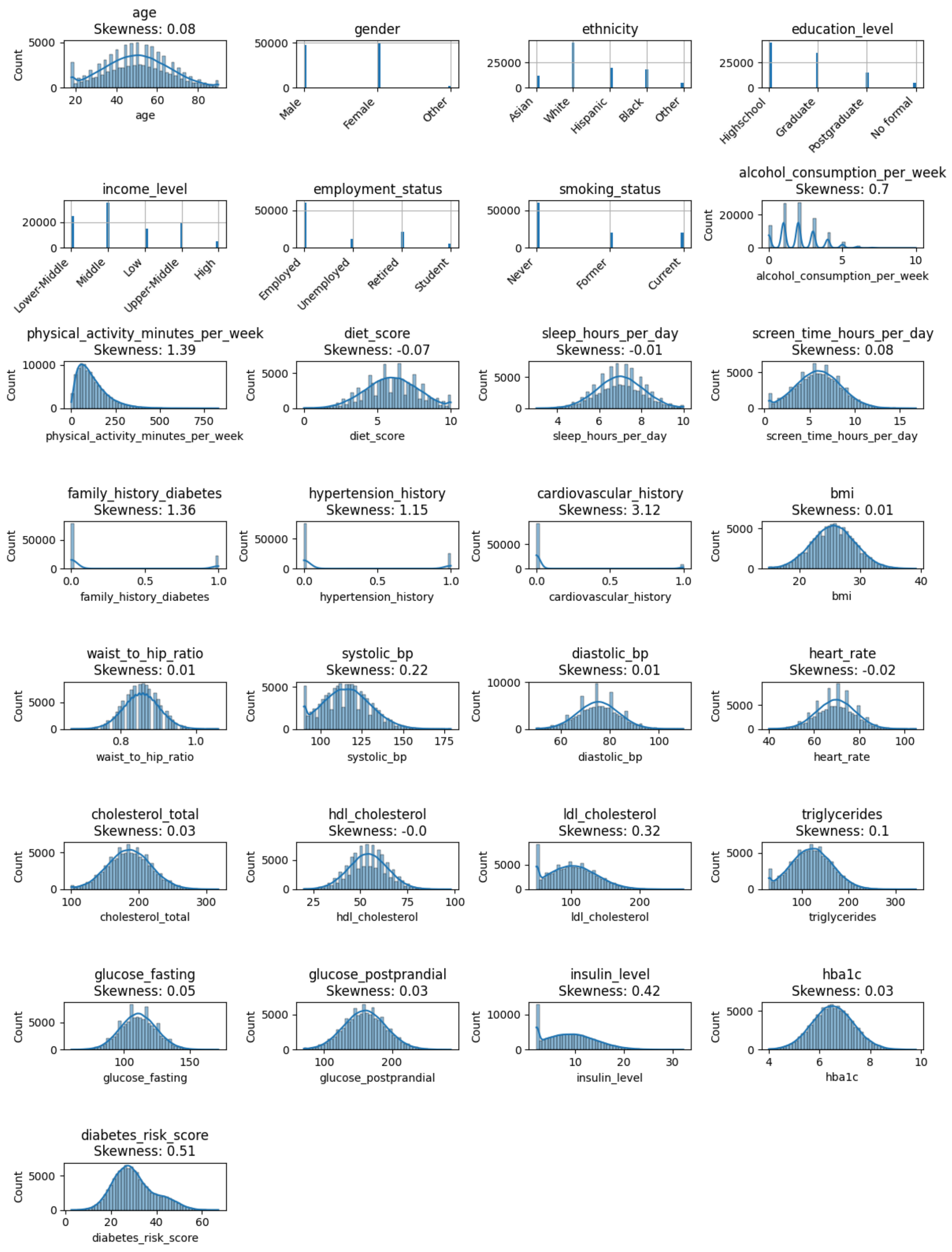
	mean	std	min	25%	50%	75
age	50.120410	15.604600	18.00	39.00	50.00	61.
alcohol_consumption_per_week	2.003670	1.417779	0.00	1.00	2.00	3.
physical_activity_minutes_per_week	118.911640	84.409662	0.00	57.00	100.00	160.
diet_score	5.994787	1.780954	0.00	4.80	6.00	7.
sleep_hours_per_day	6.997818	1.094622	3.00	6.30	7.00	7.
screen_time_hours_per_day	5.996468	2.468406	0.50	4.30	6.00	7.
family_history_diabetes	0.219410	0.413849	0.00	0.00	0.00	0.
hypertension_history	0.250800	0.433476	0.00	0.00	0.00	1.
cardiovascular_history	0.079200	0.270052	0.00	0.00	0.00	0.
bmi	25.612653	3.586705	15.00	23.20	25.60	28.
waist_to_hip_ratio	0.856078	0.046837	0.67	0.82	0.86	0.
systolic_bp	115.799610	14.284073	90.00	106.00	116.00	125.
diastolic_bp	75.232490	8.204250	50.00	70.00	75.00	81.
heart_rate	69.632870	8.371954	40.00	64.00	70.00	75.
cholesterol_total	185.978110	32.013005	100.00	164.00	186.00	208.
hdl_cholesterol	54.042790	10.267374	20.00	47.00	54.00	61.
ldl_cholesterol	103.000430	33.390256	50.00	78.00	102.00	126.
triglycerides	121.462650	43.372619	30.00	91.00	121.00	151.
glucose_fasting	111.117120	13.595610	60.00	102.00	111.00	120.
glucose_postprandial	160.035050	30.935472	70.00	139.00	160.00	181.
insulin_level	9.061242	4.954060	2.00	5.09	8.79	12.
hba1c	6.520776	0.813921	4.00	5.97	6.52	7.
diabetes_risk_score	30.222362	9.061505	2.70	23.80	29.00	35.
diagnosed_diabetes	0.599980	0.489904	0.00	0.00	1.00	1.

Looking at the summary statistics, the data appears reasonable. Age ranges from 18 to 90 with a mean around 50. BMI averages 25.6, which is slightly overweight according to standard classifications. Blood pressure values look normal with systolic averaging 116 and diastolic 75. The diabetes risk score ranges from 2.7 to 67.2 with a mean of 30.2, showing good spread in our target variable. About 60% of patients are diagnosed with diabetes, which means the dataset is somewhat imbalanced but not extremely so.

Binary history variables show that roughly 22% have family history of diabetes, 25% have hypertension history, and 8% have cardiovascular history.

Data distribution

```
In [8]: plots.plot_distributions(diabetes.drop('diagnosed_diabetes', axis=1))
```



Most numerical features show approximately symmetric distributions with skewness values close to 0. Variables like age, BMI, blood pressure readings, cholesterol levels, and glucose measurements are nearly normally distributed, which is good for many modeling approaches.

The binary variables (family history, hypertension history, cardiovascular history) show high positive skewness, which is expected since most patients don't have these conditions. Cardiovascular history has particularly high skewness at 3.12, indicating it's relatively rare in the dataset.

Physical activity shows the highest skewness among continuous variables at 1.39, with many people exercising minimally and fewer exercising heavily. Insulin levels show moderate right skew (0.42) with most patients having lower levels and some having particularly high values. Our target variable diabetes_risk_score also shows moderate right skew (0.51), suggesting some patients have particularly high risk scores that pull the distribution to the right.

The categorical variables show imbalanced distributions - gender is fairly balanced between male and female, while ethnicity is dominated by Asian and White populations. Education and income levels show concentration in the middle categories.

Looking at variable types for future encoding, we can identify several groups:

- Education level, income level, and smoking status are ordinal variables with natural ordering and should be encoded using OrdinalEncoder. - Employment status, gender, and ethnicity are nominal categories without inherent ordering and will need OneHotEncoder.
- The binary history variables (family history, hypertension, cardiovascular) are already encoded as 0/1, though it's worth noting that gender should not be treated as strictly binary.
- Alcohol consumption per week can be treated as a continuous numeric variable given its distribution.

Looking for outliers, features that are logical or input errors.

In [9]:
```python
# blood pressure
impossible_bp = diabetes[
    (diabetes['systolic_bp'] <= diabetes['diastolic_bp'])
    | (diabetes['systolic_bp'] < 60)
    | (diabetes['systolic_bp'] > 300)
]
print(
    f'BP (systolic must be higher than diastolic, out of possible range): '
    f'{len(impossible_bp)} rows.'
)

# cholesterol
cholesterol_error = diabetes[
    diabetes['cholesterol_total']
    < (diabetes['hdl_cholesterol'] + diabetes['ldl_cholesterol']) - 20
]
```

```
print(f'Cholesterol (Total < HDL+LDL): {len(cholesterol_error)} rows.')

# screen time, sleep hours, age
sleep_error = diabetes[
    (diabetes['sleep_hours_per_day'] <= 0) | (diabetes['sleep_hours_per_day'
]

age_error = diabetes[(diabetes['age'] < 0) | (diabetes['age'] > 110)]

screentime_error = diabetes[
    (diabetes['screen_time_hours_per_day'] < 0)
    | (diabetes['screen_time_hours_per_day'] > 24)
]

print(f'Sleep hours per day error(<=0 or >24h): {len(sleep_error)} rows.')
print(f'Age error: {len(age_error)} rows.')
print(f'Screen time error: {len(screentime_error)} rows.')

# WHR (Waist to hip ratio)
whr_error = diabetes[
    (diabetes['waist_to_hip_ratio'] < 0.5) | (diabetes['waist_to_hip_ratio']
]
print(f'WHR error: {len(whr_error)} rows.')

# lab errors (impossible zeros)
lab_cols = ['insulin_level', 'cholesterol_total', 'triglycerides']
for col in lab_cols:
    zeros = len(diabetes[diabetes[col] == 0])
    if zeros > 0:
        print(f'ERROR in {col}: {zeros}')
```

```
BP (systolic must be higher than diastolic, out of possible range): 154 rows.
Cholesterol (Total < HDL+LDL): 36 rows.
Sleep hours per day error(<=0 or >24h): 0 rows.
Age error: 0 rows.
Screen time error: 0 rows.
WHR error: 0 rows.
```

We identified 190 records with data quality issues: 154 rows with impossible blood pressure readings (systolic ≤ diastolic or out of physiological range) and 36 rows where total cholesterol was significantly lower than the sum of HDL and LDL cholesterol, indicating measurement or recording errors that need to be removed before modeling.

Data Cleaning - Removing Invalid Records

In [10]:
```
# Deleting
index_to_drop = (
    set(impossible_bp.index)
    | set(cholesterol_error.index)
    | set(sleep_error.index)
    | set(age_error.index)
    | set(screentime_error.index)
    | set(whr_error.index)
)
```

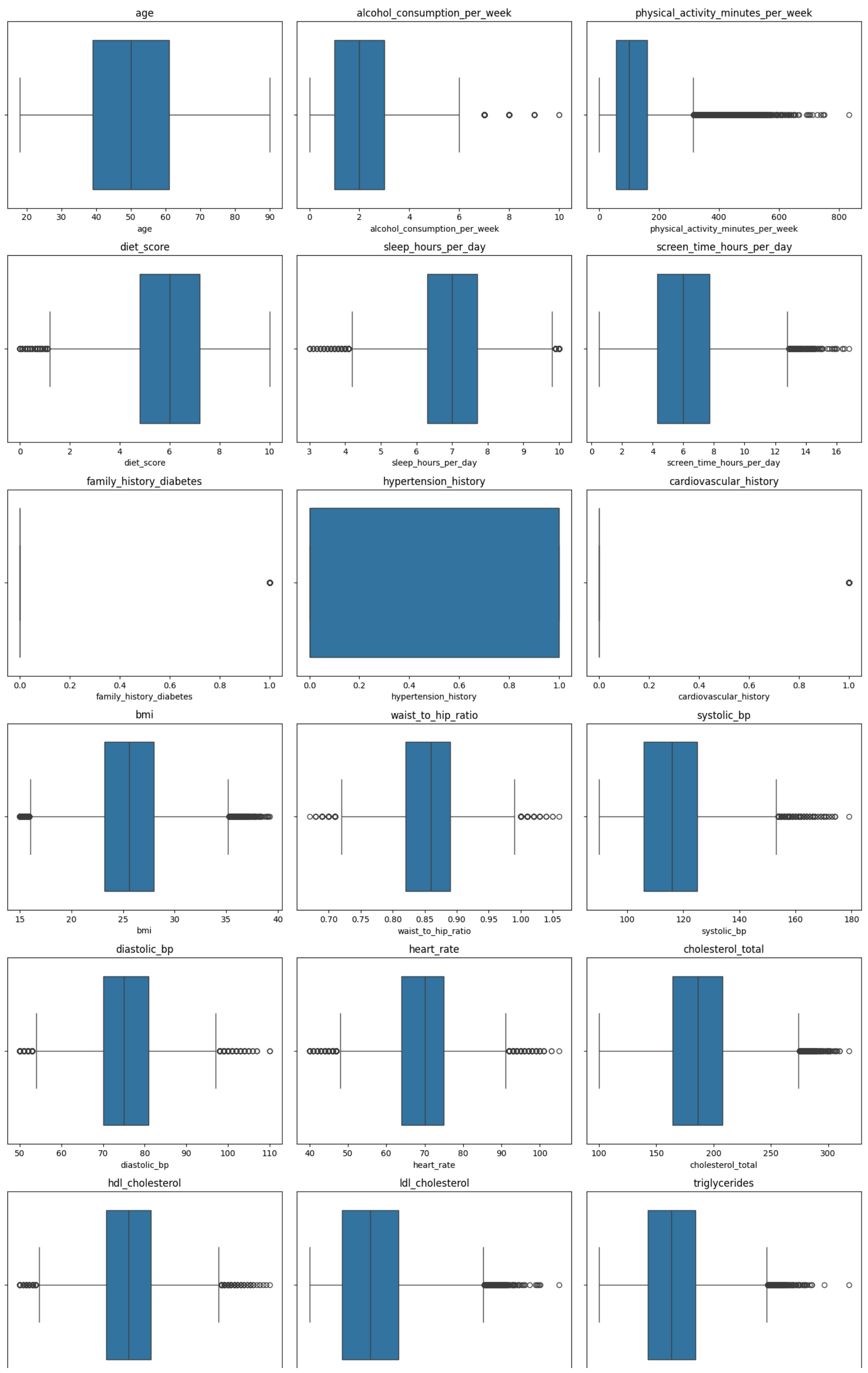
```python
diabetes_clean = diabetes.drop(index=list(index_to_drop)).copy()

print(f'Data rows before: {len(diabetes)}, after: {len(diabetes_clean)}')
```

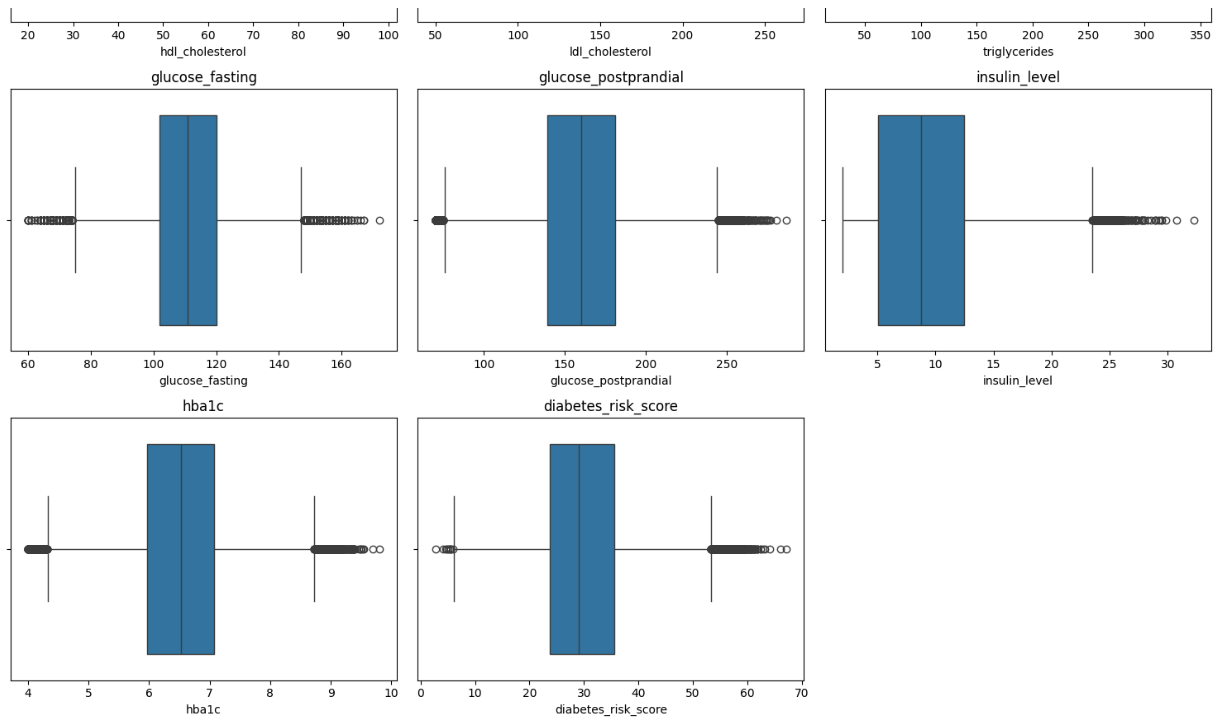
```
Data rows before: 100000, after: 99810
```

After removing 190 invalid records (0.19% of the dataset), we retained 99,810 clean records for analysis and modeling.

Checking for outliers (Boxplots)

```
In [11]: plots.plot_numerical_boxplots(diabetes_clean.drop(['diagnosed_diabetes'], ax
```


age
alcohol_consumption_per_week
physical_activity_minutes_per_week
diet_score
sleep_hours_per_day
screen_time_hours_per_day
family_history_diabetes
hypertension_history
cardiovascular_history
bmi
waist_to_hip_ratio
systolic_bp
diastolic_bp
heart_rate
cholesterol_total
hdl_cholesterol
ldl_cholesterol
triglycerides

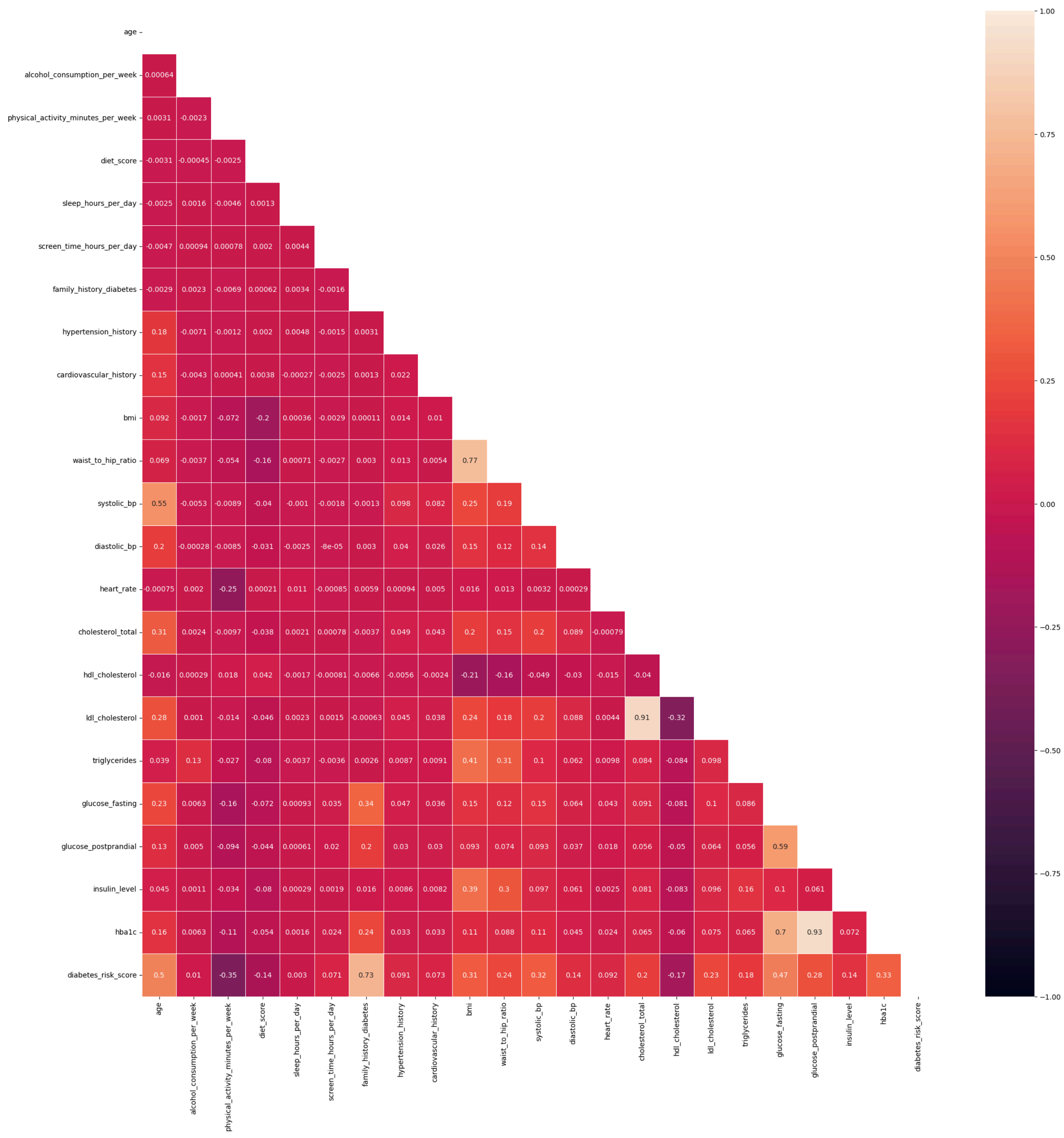


After removing the impossible values, the remaining outliers visible in the boxplots appear to represent natural variation rather than data errors. Variables like insulin level, triglycerides, and postprandial glucose show significant outliers on the high end, likely representing patients with metabolic syndrome or advanced diabetes. We've decided not to remove these outliers as they represent valid patient profiles good for the model to learn, and will instead handle them during preprocessing with scaling methods.

Correlation

In [12]:
```python
corr = diabetes_clean.drop(['diagnosed_diabetes'], axis=1).corr(numeric_only
mask = np.triu(np.ones_like(corr, dtype=bool))

plt.figure(figsize=(25, 25))
sns.heatmap(
    corr,
    annot=True,
    vmin=-1,
    vmax=1,
    mask=mask,
    linewidths=0.5,
)
```

Out[12]:
```
<Axes: >
```



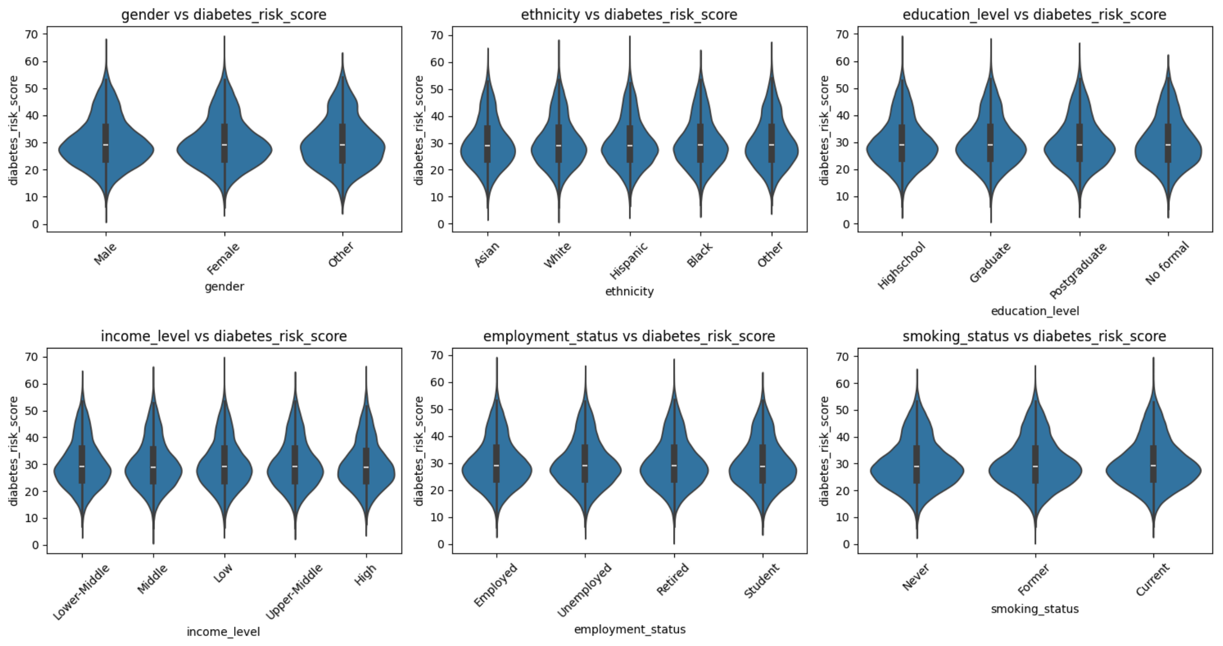
The strongest predictors of diabetes risk score are family history of diabetes (r=0.73), age (r=0.50), and fasting glucose (r=0.47). Physical activity shows a moderate negative correlation (r=-0.35), suggesting it's protective against diabetes risk. Interestingly, lifestyle factors like sleep hours and alcohol consumption show negligible correlations with the target variable.

We identified several concerning multicollinearity issues among predictor variables. Glucose postprandial and HbA1c are extremely highly correlated (r=0.93), as are total cholesterol and LDL cholesterol (r=0.91). Fasting glucose correlates strongly with both HbA1c (r=0.70) and postprandial glucose (r=0.59). BMI and waist-to-hip ratio also show high correlation (r=0.77). These redundancies mean that including all these variables together in linear models could lead to unstable coefficient estimates and inflated standard errors.

For modeling, we have several options to handle this multicollinearity. For linear models like Ridge or Lasso regression, we can keep all variables since regularization naturally handles correlated features. For standard linear regression or models sensitive to multicollinearity, we should consider dropping one variable from each highly correlated pair. Alternatively, tree-based models like Random Forest or XGBoost are naturally resistant to multicollinearity and can use all features without issues. We could also apply PCA to create uncorrelated components, though this would sacrifice interpretability.

Distribution of target value for different categories for each categorical value

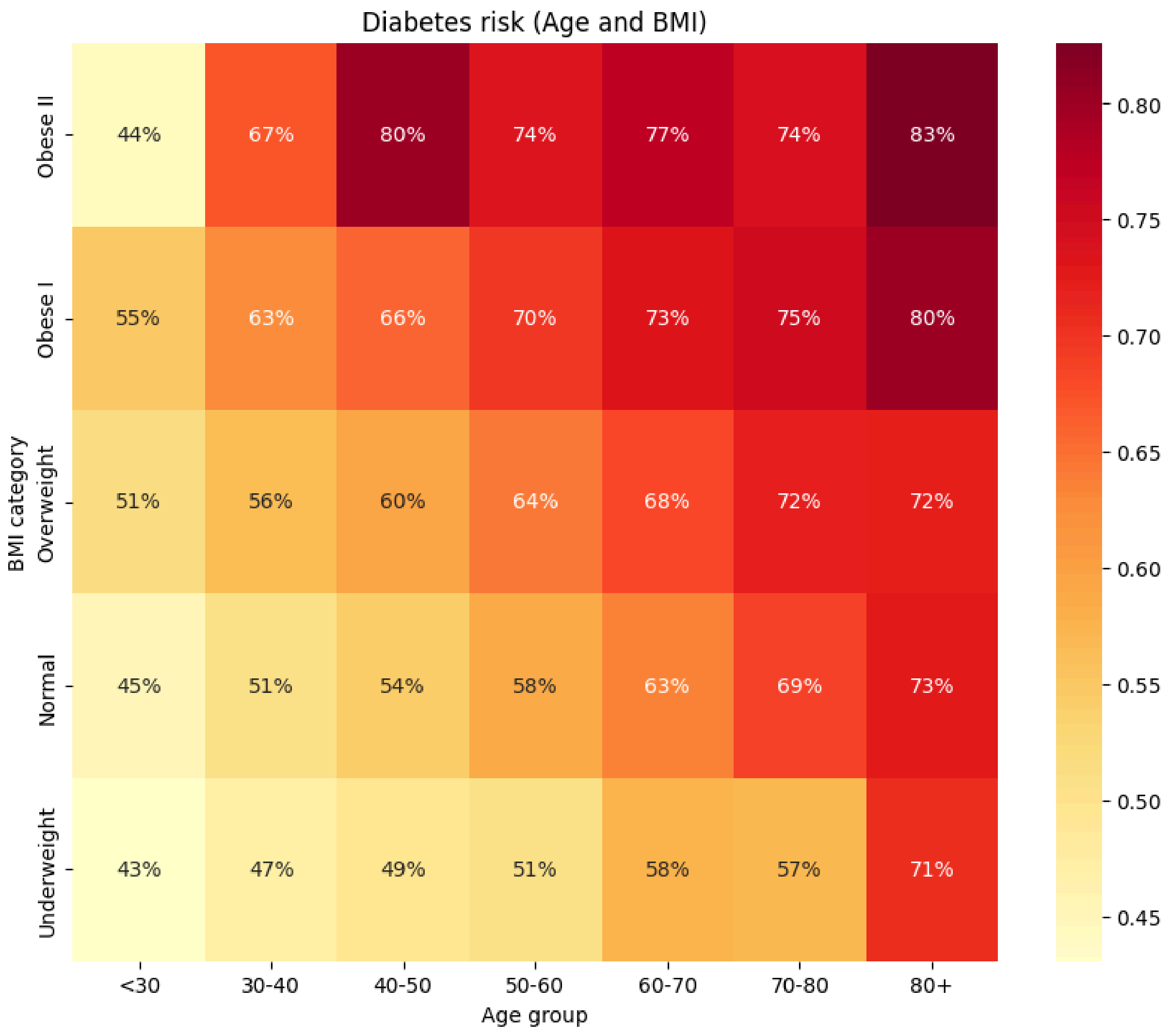
In [13]:
```
plots.plot_target_vs_category(diabetes_clean, 'diabetes_risk_score', n_cols=
```



The violin plots reveal that categorical variables like gender, ethnicity, education, income, employment, and smoking status show nearly identical diabetes risk score distributions across all categories, suggesting these demographic factors have minimal individual impact compared to continuous health metrics.

Diabetes Risk Analysis - Age and BMI

In [14]:
```
# grouping by age and BMI
bmi_bins = [0, 18.5, 25, 30, 35, 40, np.inf]
bmi_labels = [
    'Underweight',
    'Normal',
    'Overweight',
    'Obese I',
    'Obese II',
    'Obese III+',
]

age_bins = [0, 30, 40, 50, 60, 70, 80, np.inf]
age_labels = ['<30', '30-40', '40-50', '50-60', '60-70', '70-80', '80+']

diabetes_clean['bmi_group'] = pd.cut(
    diabetes_clean['bmi'], bins=bmi_bins, labels=bmi_labels
)
diabetes_clean['age_group'] = pd.cut(
    diabetes_clean['age'], bins=age_bins, labels=age_labels
)

#
risk_matrix = diabetes_clean.pivot_table(
    index='bmi_group',
    columns='age_group',
    values='diagnosed_diabetes',
    aggfunc='mean',
    observed=False,
)

plt.figure(figsize=(10, 8))
sns.heatmap(risk_matrix, annot=True, fmt='.0%', cmap='YlOrRd')
plt.title('Diabetes risk (Age and BMI)')
plt.ylabel('BMI category')
plt.xlabel('Age group')
plt.gca().invert_yaxis()
plt.show()
```



The heatmap shows a clear diagonal pattern where diabetes risk increases with both age and BMI, confirming their combined effect on disease prevalence. The highest risk appears in older age groups with higher obesity levels, reaching 80-83% in the oldest and most obese categories. Interestingly, even young individuals with normal or low BMI show relatively high diabetes rates (43-51%), suggesting the dataset may include Type 1 diabetes cases or other risk factors beyond age and BMI alone.

Summary

Based on the exploratory findings, we hypothesize that the `diabetes_risk_score` is primarily a function of biological and genetic markers such as age, BMI, and family history, significantly outweighing the influence of demographic variables. We further postulate that while elevated glucose metrics will display a strong positive correlation with the risk score, physical activity will emerge as a critical negative predictor, validating its protective role in the regression model.