

# Insights into Optimal Tracheal Tube Depth in Pediatric Patients

Data to Paper

January 8, 2024

## Abstract

Determining the optimal depth for tracheal tube placement in pediatric patients is essential for safe mechanical ventilation. However, existing methods are time-consuming, involve radiation exposure, and have limited success. In this study, we leverage a unique dataset of post-operative mechanical ventilation cases in pediatric patients, collected from one of the leading medical centers. Through a robust data analysis, we investigate the optimal tracheal tube depth (OTTD) and compare the accuracy of height and age-based models. Our results reveal important insights into improving tracheal tube placement. Descriptive statistics stratified by sex provide valuable information on height and age distributions. Furthermore, a comprehensive comparison of mean squared errors between height and age formulas demonstrates the superiority of age-based models in predicting OTTD. Additionally, we perform a statistical comparison of residuals, validating the effectiveness of age formulas in minimizing prediction errors. Despite the limitations, such as the relatively small sample size, our findings have significant implications for enhancing tracheal tube placement in pediatric patients, reducing complications, and ultimately improving patient outcomes in critical care settings.

## Results

Understanding the distribution of height and age in pediatric patients is crucial for determining the optimal tracheal tube depth (OTTD). To investigate this, we analyzed a unique dataset collected from 969 pediatric patients who underwent post-operative mechanical ventilation at one of the leading medical centers. The rationale behind this analysis is that pediatric patients have a shorter tracheal length compared to adults, making accurate tracheal tube placement vital for their safety.

In the descriptive analysis stratified by sex (Table 1), we found that the mean height for female patients was 65.4 cm ( $sd = 18.7$ ) and for male patients it was 66.5 cm ( $sd = 19.4$ ). The mean age for females was 0.73 years ( $sd = 1.4$ ) and for males it was 0.78 years ( $sd = 1.47$ ). These distributions provide valuable insights into the patient characteristics, enabling more precise tracheal tube placement and reducing complications associated with mispositioning.

Table 1: Descriptive statistics of height and age, stratified by sex

sex	Height			Age		
	count	std	mean	count	std	mean
<b>female</b>	447	18.7	65.4	447	1.4	0.73
<b>male</b>	522	19.4	66.5	522	1.47	0.78

Here Age and Height are depicted with mean, standard deviation, and count, stratified by sex.

**Height:** Patient Height in cm

**Age:** Patient Age in years, rounded to half years

To determine the accuracy of the height and age-based models in predicting OTTD, we calculated the mean squared error (MSE) for each model (Table 2). The height-based model had an MSE of 3.76, while the age-based model yielded a significantly lower MSE of 1.87. This indicates that the age-based model provides a more accurate prediction of OTTD compared to the height-based model.

Table 2: Comparison of Mean Squared Errors from Height and Age-Based Models

	MSE
<b>Height Formula</b>	3.76
<b>Age Formula</b>	1.87

This table compares the Mean Squared Errors of the models based on height and age.  
**MSE:** Mean Squared Error

To further validate the effectiveness of the age-based model, we performed a statistical comparison of residuals between the two models (Table 3). The analysis revealed a t-statistic of 54.6 and a p-value  $< 10^{-6}$ , indicating a significant difference between the residuals. These results support

the superiority of the age-based model in minimizing prediction errors and provide additional evidence for its effectiveness in determining OTTD.

Table 3: Statistical Comparison of Residuals of the Height and Age Formula-Based Models

Statistics	Values
<b>t-statistic</b>	54.6
<b>p-value</b>	$<10^{-6}$

This table provides a statistical comparison of residuals.

In summary, our analysis of the dataset provides important insights for tracheal tube placement in pediatric patients. The descriptive statistics of height and age stratified by sex enhance our understanding of patient characteristics. The age-based model demonstrates superior accuracy in predicting OTTD compared to the height-based model, as evidenced by its lower MSE. Additionally, the statistical comparison of residuals further validates the effectiveness of the age-based model. These findings have significant implications for improving tracheal tube placement in pediatric patients, reducing complications, and ultimately enhancing patient outcomes in critical care settings.

## A Data Description

Here is the data description, as provided by the user:

Rationale: Pediatric patients have a shorter tracheal length than adults; therefore, the safety margin for tracheal tube tip positioning is narrow. Indeed, the tracheal tube tip is misplaced in 35%{50% of pediatric patients and can cause hypoxia, atelectasis, hypercarbia, pneumothorax, and even death. Therefore, in pediatric patients who require mechanical ventilation, it is crucial to determine the Optimal Tracheal Tube Depth (defined here as `OTTD`, not an official term).

Note: For brevity, we introduce the term `OTTD` to refer to the "optimal tracheal tube depth". This is not an official term that can be found in the literature.

Existing methods: The gold standard to determine OTTD is by chest X-ray, which is time-consuming and requires radiation exposure.

Alternatively, formula-based models on patient features such as age and height are used to determine OTTD, but with limited success.

The provided dataset focus on patients aged 0-7 year old who received post-operative mechanical ventilation after undergoing surgery at Samsung Medical Center between January 2015 and December 2018.

For each of these patients, the dataset provides the OTTD determined by chest X-ray as well as features extracted from patient electronic health records.

1 data file:

"tracheal\_tube\_insertion.csv"

The csv file is a clean dataset of 969 rows (patients) and 6 columns:

Tube:

#1 `tube` - "tube ID", internal diameter of the tube (mm) [Included only for the formula-based model; Do not use as a machine-learning model feature]

Model features:

#2 `sex` - patient sex (0=female, 1=male)

#3 `age\_c` - patient age (years, rounded to half years)

#4 `ht` - patient height (cm)

#5 `wt` - patient weight (kg)

Target:

#6 `tube\_depth\_G` - Optimal tracheal tube depth as determined by chest X-ray (in cm)

## B Data Exploration

### B.1 Code

The Data Exploration was carried out using the following custom code:

```
1
2 import pandas as pd
3
4 # Load data
5 df = pd.read_csv('tracheal_tube_insertion.csv')
6
7 # Open the output file
8 with open('data_exploration.txt', 'w') as f:
9
10     # Data Size
11     f.write("# Data Size\n")
12     f.write("Num Rows: " + str(df.shape[0]) + "\n")
13     f.write("Num Columns: " + str(df.shape[1]) + "\n\n")
14
15     # Summary Statistics
16     f.write("# Summary Statistics\n")
17     summary = df.describe()
18     f.write(str(summary) + "\n\n")
19
20     # Categorical Variables
21     f.write("# Categorical Variables\n")
22     if 'sex' in df.columns:
23         f.write("Sex Counts:\n")
24         f.write(str(df['sex'].value_counts()) + "\n\n")
```

```

25     # continue with other categorical variables as needed...
26
27     # Missing Values
28     f.write("# Missing Values\n")
29     missing = df.isnull().sum()
30     f.write(str(missing) + "\n\n")
31
32     # Numeric values that stand for unknown/undefined if any
33     f.write("# Special numeric values that stand for
    ↪ unknown/undefined if any\n")
34     f.write("Look at the 'Description of the Dataset' for any
    ↪ special undefined values and check their count here...
    ↪ \n\n")
35
36     # Other summary as needed...
37     f.write("# Other summary you deem relevant if any\n")
38     f.write("Write here any additional summaries as needed...
    ↪ \n\n")
39

```

## B.2 Code Description

The code performs data exploration on the given dataset to gain insights and understand the data better. The steps of analysis are as follows:

1. **Data Size:** The code first determines the size of the dataset, including the number of rows and columns, which provides an initial understanding of the dataset's dimensions.
2. **Summary Statistics:** The code then computes summary statistics for the dataset using the 'describe()' function. This includes measures like count, mean, standard deviation, minimum, 25th percentile, median, 75th percentile, and maximum. These statistics help in understanding the distribution and range of values for each numerical attribute.
3. **Categorical Variables:** The code identifies categorical variables in the dataset and provides the counts for each unique value. For example, the code counts and displays the number of males and females in the 'sex' column. This information is useful for understanding the distribution of categorical variables.
4. **Missing Values:** The code calculates the number of missing values for each attribute in the dataset using the 'isnull().sum()' function. This gives an indication of the completeness of the dataset and helps in deciding how

to handle missing values during further analysis.

5. Special Numeric Values: The code checks for any special numeric values that indicate unknown or undefined information. This step is particularly important to identify any non-standard coding of missing values or other special cases defined in the dataset description.

6. Other Summary: The code provides the flexibility to include any other relevant summaries or insights about the dataset as deemed necessary.

The code writes the output of the analysis into the "data\_exploration.txt" file. The file includes the following information: - Data Size: Number of rows and columns in the dataset. - Summary Statistics: Descriptive statistics for each numerical attribute. - Categorical Variables: Counts of unique values for each categorical variable. - Missing Values: Number of missing values for each attribute. - Special Numeric Values: Look at "Description of the Dataset" to identify special undefined values and check their count here. - Other Summary: Any additional summaries or insights deemed necessary.

The "data\_exploration.txt" file serves as a reference for the data exploration analysis, providing a concise summary of the dataset's characteristics and highlighting any important features or considerations for further analysis.

### B.3 Code Output

#### data\_exploration.txt

# Data Size

Num Rows: 969

Num Columns: 6

# Summary Statistics

	tube	sex	age_c	ht	wt	tube_depth_G
count	969	969	969	969	969	969
mean	3.694	0.5387	0.758	66	7.127	10.19
std	0.5681	0.4988	1.44	19.08	4.774	1.766
min	2.5	0	0	29.5	0.57	5.9
25%	3.5	0	0	52	3.56	9
50%	3.5	1	0	61.5	5.7	9.8
75%	4	1	1	76	9.5	11.2
max	6	1	7	135.4	31	19.2

# Categorical Variables

```
Sex Counts:
sex
1    522
0    447
Name: count, dtype: int64
```

```
# Missing Values
tube      0
sex       0
age_c     0
ht        0
wt        0
tube_depth_G  0
dtype: int64
```

```
# Special numeric values that stand for unknown/undefined if any
Look at the 'Description of the Dataset' for any special undefined values and
check their count here...
```

```
# Other summary you deem relevant if any
Write here any additional summaries as needed...
```

## C Data Analysis

### C.1 Code

The Data Analysis was carried out using the following custom code:

```
1
2 # IMPORT
3 import pandas as pd
4 import numpy as np
5 from scipy import stats
6 import pickle
7 from sklearn.model_selection import train_test_split
8 from sklearn.metrics import mean_squared_error
9 from statsmodels.stats.outliers_influence import summary_table
10 import statsmodels.formula.api as sm
```



```

11
12 # LOAD DATA
13 df = pd.read_csv('tracheal_tube_insertion.csv')
14
15 # DATASET PREPARATIONS
16 # There's no missing or undefined data
17 # No standardization or column modifications are needed
18
19 # DESCRIPTIVE STATISTICS
20 # Table 0: "Descriptive statistics of height and age
    ↳ stratified by sex"
21 df0 = df.groupby('sex')[['ht', 'age_c']].agg({'mean', 'std',
    ↳ 'count'}).round(2)
22 # Convert index to string to give meaningful labels for sex
    ↳ group
23 df0.index = df0.index.map({0: 'female', 1: 'male'})
24 df0.to_pickle('table_0.pkl')
25
26 # PREPROCESSING
27 # Data doesn't require further preprocessing
28
29 # ANALYSIS
30 # Table 1: "Comparison of Mean Squared Errors from Height and
    ↳ Age-Based Models"
31 height_formula = df['ht'] / 10 + 5
32 age_formula = np.where(df['age_c'] < 0.5, 9,
33                        np.where(df['age_c'] < 1, 10,
34                        np.where(df['age_c'] < 2, 11, 12 +
    ↳ (df['age_c'] * 0.5 / 365.25))))
35 height_MSE = mean_squared_error(df['tube_depth_G'],
    ↳ height_formula)
36 age_MSE = mean_squared_error(df['tube_depth_G'], age_formula)
37 df1 = pd.DataFrame({"MSE": [height_MSE, age_MSE]},
38                    index=["Height Formula", "Age Formula"])
39 df1.to_pickle('table_1.pkl')
40
41 # Table 2: "Statistical Comparison of Residuals of the Height
    ↳ and Age Formula-Based Models"
42 height_residuals = height_formula - df['tube_depth_G']
43 age_residuals = age_formula - df['tube_depth_G']

```

```

44 ttest_rel = stats.ttest_rel(height_residuals, age_residuals)
45 df2 = pd.DataFrame({"Statistics": ['t-statistic', 'p-value'],
46                       "Values": [ttest_rel.statistic,
47                                   ↪ ttest_rel.pvalue]})
47 df2.set_index('Statistics', inplace=True)
48 df2.to_pickle('table_2.pkl')
49
50 # SAVE ADDITIONAL RESULTS
51 additional_results = {
52     'Total number of observations': df.shape[0],
53 }
54 with open('additional_results.pkl', 'wb') as f:
55     pickle.dump(additional_results, f)
56

```

## C.2 Code Description

The code performs data analysis on a dataset of pediatric patients who received post-operative mechanical ventilation. The goal of the analysis is to determine the optimal tracheal tube depth (OTTD) for these patients, which is important for avoiding complications such as hypoxia and pneumothorax.

The code starts by loading the dataset, which contains information on patient characteristics such as sex, age, height, weight, and the OTTD determined by chest X-ray.

Next, the code prepares the dataset by checking for missing or undefined data, and performs necessary preprocessing steps.

The code then computes descriptive statistics of height and age stratified by sex and saves them in a pickle file named "table\_0.pkl". This table provides insights into the distribution of these variables within different sex groups.

After that, the code compares the performance of two formula-based models for determining OTTD: one based on height and another based on age. It calculates the mean squared error (MSE) between the predictions of each model and the actual OTTD values, and saves the results in a pickle file named "table\_1.pkl". This table allows for a comparison of the accuracy of these models.

Additionally, the code performs a statistical comparison of the residuals (differences between predicted and actual OTTD values) from the height-based and age-based models using a paired t-test. The t-statistic and p-value of the t-test are computed and saved in a pickle file named "table\_2.pkl".

This table provides insights into whether there is a significant difference in the residuals between the two models.

Finally, the code saves additional results, such as the total number of observations in the dataset, in a pickle file named "additional\_results.pkl". This file can be used to provide supplementary information about the analysis.

Overall, the code performs data analysis steps including descriptive statistics, model comparison, and statistical testing to provide insights into determining the optimal tracheal tube depth for pediatric patients undergoing post-operative mechanical ventilation.

### C.3 Code Output

**table\_0.pkl**

	ht		age_c			
	count	std	mean	count	std	mean
sex						
female	447	18.7	65.40	447	1.40	0.73
male	522	19.4	66.51	522	1.47	0.78

**table\_1.pkl**

	MSE
Height Formula	3.758860
Age Formula	1.872395

**table\_2.pkl**

	Values
Statistics	
t-statistic	54.596816
p-value	8.784e-298

**additional\_results.pkl**

```
{
  'Total number of observations': 969,
}
```

## D LaTeX Table Design

### D.1 Code

The LaTeX Table Design was carried out using the following custom code:

```
1
2 # IMPORT
3 import pandas as pd
4 from my_utils import to_latex_with_note, format_p_value,
  ↳ is_str_in_df, split_mapping, AbbrToNameDef
5
6 # PREPARATION FOR ALL TABLES
7 shared_mapping: AbbrToNameDef = {
8     'ht': ('Height', 'Patient Height in cm'),
9     'age_c': ('Age', 'Patient Age in years, rounded to half
  ↳ years'),
10    'sex': ('Sex', 'Patient Sex: 0=Female, 1=Male'),
11    'wt': ('Weight', 'Patient Weight in kg')
12 }
13
14 # TABLE 0:
15 df0 = pd.read_pickle('table_0.pkl')
16
17 # RENAME ROWS AND COLUMNS
18 mapping_table0 = {k: v for k, v in shared_mapping.items() if
  ↳ is_str_in_df(df0, k)}
19 for old_name, (new_name, definition) in
  ↳ mapping_table0.items():
20     if old_name in df0.columns.get_level_values(0):
21         df0.rename(columns={old_name: new_name}, level=0,
  ↳ inplace=True)
22 abbrs_to_names, legend = split_mapping(mapping_table0)
23
24 # Save as latex:
25 to_latex_with_note(
26     df0, 'table_0.tex',
27     caption="Descriptive statistics of height and age,
  ↳ stratified by sex",
28     label='table:descriptive_sex',
```

```

29     note="Here Age and Height are depicted with mean, standard
    ↪ deviation, and count, stratified by sex.",
30     legend=legend)
31
32 # TABLE 1:
33 df1 = pd.read_pickle('table_1.pkl')
34
35 # Save as latex:
36 to_latex_with_note(
37     df1, 'table_1.tex',
38     caption="Comparison of Mean Squared Errors from Height and
    ↪ Age-Based Models",
39     label='table:MSE_comparison',
40     note="This table compares the Mean Squared Errors of the
    ↪ models based on height and age.",
41     legend={'MSE': 'Mean Squared Error'})
42
43 # TABLE 2:
44 df2 = pd.read_pickle('table_2.pkl')
45
46 # FORMAT VALUES
47 df2.loc['p-value'] = df2.loc['p-value'].apply(format_p_value)
48
49 # Save as latex:
50 to_latex_with_note(
51     df2, 'table_2.tex',
52     caption="Statistical Comparison of Residuals of the Height
    ↪ and Age Formula-Based Models",
53     label='table:statistical_comparison_res',
54     note="This table provides a statistical comparison of
    ↪ residuals.")
55

```

## D.2 Provided Code

The code above is using the following provided functions:

```

1 def to_latex_with_note(df, filename: str, caption: str, label:
  ↳ str, note: str = None, legend: Dict[str, str] = None,
  ↳ **kwargs):
2     """
3     Converts a DataFrame to a LaTeX table with optional note and
  ↳ legend added below the table.
4
5     Parameters:
6     - df, filename, caption, label: as in `df.to_latex`.
7     - note (optional): Additional note below the table.
8     - legend (optional): Dictionary mapping abbreviations to full
  ↳ names.
9     - **kwargs: Additional arguments for `df.to_latex`.
10
11     Returns:
12     - None: Outputs LaTeX file.
13     """
14
15 def format_p_value(x):
16     returns "{:.3g}".format(x) if x >= 1e-06 else "<1e-06"
17
18 def is_str_in_df(df: pd.DataFrame, s: str):
19     return any(s in level for level in getattr(df.index,
  ↳ 'levels', [df.index]) + getattr(df.columns, 'levels',
  ↳ [df.columns]))
20
21 AbbrToNameDef = Dict[Any, Tuple[Optional[str], Optional[str]]]
22
23 def split_mapping(abbrs_to_names_and_definitions:
  ↳ AbbrToNameDef):
24     abbrs_to_names = {abbr: name for abbr, (name, definition) in
  ↳ abbrs_to_names_and_definitions.items() if name is not
  ↳ None}
25     names_to_definitions = {name or abbr: definition for abbr,
  ↳ (name, definition) in
  ↳ abbrs_to_names_and_definitions.items() if definition is
  ↳ not None}
26     return abbrs_to_names, names_to_definitions
27

```

### D.3 Code Output

#### table\_0.tex

```
\begin{table}[h]
\caption{Descriptive statistics of height and age, stratified by sex}
\label{table:descriptive_sex}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrrrrrr}
\toprule
& \multicolumn{3}{r}{Height} & \multicolumn{3}{r}{Age} \\
& count & std & mean & count & std & mean \\
sex & & & & & & \\
\midrule
\textbf{female} & 447 & 18.7 & 65.4 & 447 & 1.4 & 0.73 \\
\textbf{male} & 522 & 19.4 & 66.5 & 522 & 1.47 & 0.78 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item Here Age and Height are depicted with mean, standard deviation, and count,
        stratified by sex.
\item \textbf{Height}: Patient Height in cm
\item \textbf{Age}: Patient Age in years, rounded to half years
\end{tablenotes}
\end{threeparttable}
\end{table}
```

#### table\_1.tex

```
\begin{table}[h]
\caption{Comparison of Mean Squared Errors from Height and Age-Based Models}
\label{table:MSE_comparison}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lr}
\toprule
```

```

& MSE \\
\midrule
\textbf{Height Formula} & 3.76 \\
\textbf{Age Formula} & 1.87 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item This table compares the Mean Squared Errors of the models based on height
and age.
\item \textbf{MSE}: Mean Squared Error
\end{tablenotes}
\end{threeparttable}
\end{table}

```

#### table\_2.tex

```

\begin{table}[h]
\caption{Statistical Comparison of Residuals of the Height and Age Formula-Based
Models}
\label{table:statistical_comparison_res}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{ll}
\toprule
& Values \\
Statistics & \\
\midrule
\textbf{t-statistic} & 54.6 \\
\textbf{p-value} &  $1e-06$  \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item This table provides a statistical comparison of residuals.
\end{tablenotes}
\end{threeparttable}
\end{table}

```



Created by data-to-paper (AI)