A Statistical Approach to Predicting Emphysema Percentage

Connecting Performance with Contract Value

Benjamin De La Torre Martinez

STAT 610
Fall 2025
2020-12-08

Executive Summary

Emphysema is a progressive lung disease that damages alveoli (microscopic air sacs fundamental to the respiratory system). While the precise figure is uncertain, an estimated 1.6% of adults had a diagnosis of emphysema. This study thus attempts to identify the set of combined factors that are most strongly associated with the percentage of emphysema an indiviual has in their lungs. After completing analysis through linear regression we can observe that the variables that are most correlation with percentage of emphysema are those related to the health of one's respiratory system. More specifically prior diagnoses of copd and emphysema. Additionally, the health related metrics included in our model, such as smoking status, average cigarettes smoked per day, smoking duration, functional residual capacity, inspiratory and expiratory mean attenuation, gas trapping percentage, the FEV1/FVC ratio, and FVC, all contribute meaningful information about a patient's respiratory condition. These variables capture both physiological function and exposure-related effects, helping to further explain variation in emphysema percentage beyond prior diagnoses alone. Or in other words, the worse an individual's respiratory health is, the higher the percentage of emphysema tends to be.

Introduction

Most sources identify smoking and long-term exposure to second-hand smoke as the leading causes of emphysema. Rather than proving causation, this study focuses on a related question: Which variables show a significant correlation with emphysema percentage? To investigate this, we examine several indicators of general health (BMI, heart rate, blood pressure) and respiratory health. Our hypothesis is that emphysema severity is associated with negative respiratory conditions (such as COPD and prior emphysema diagnoses), smoking quantity and duration, and lung-function metrics, though the specific contributions of each remain uncertain. We apply multivariate linear regression to predict continuous emphysema percentages (0–100%) using both continuous and categorical variables. This method is appropriate for our data structure, and diagnostic checks will confirm whether the OLS assumptions are satisfied. While other analytical approaches exist, linear regression is chosen for its interpretability and its ability to highlight clear relationships between predictors and emphysema percentage. Although regression assumes mainly additive effects-a limitation given the complexity of respiratory interactions-we prioritize interpretability over predictive accuracy, making regression a justified choice for this analysis.

Exploratory Data Analysis

Our study begins by examining our data. By doing so, we get to understand the structure, distribution, range and patterns of out data. Ensuring that our analysis is consistent and account for the variability of our data.

A. Summary of our Data

Our dataset contains measurements of 5747 different individuals containing 35 predictors. One disclaimer about our data is that our study seems to be highly biased as subjects are either Caucasian or African American. 14% of variables are missing.

Table 1: Data summary

Name	df_copd
Number of rows	5747
Number of columns	35
Column type frequency:	
character	2
factor	11
numeric	22
Group variables	None

Based on the distributional profiles of all variables, no anomalies are evident. The variables exhibit mild right skewness, mild left skewness, or near-normal distributions. With the exception of visit_age which roughly resembles a uniform distribution. For more detail, see appendix.
Disclaimer: Not all variables are shown, only a subset.

variable	unique	missing	mean	sd	min	median	max
visit_age	392	0	59.75	8.688	39	59.5	85
bmi	2052	0	29.08	6.141	12.67	28.2	64.1
CigPerDaySmokAvg	47	0	23.78	11.42	0	20	99
SmokStartAge	45	0	16.74	4.821	0	16	50
pct_gastrapping	4089	1650	19.81	17.72	0.04412	13.63	81.27

B. Visualizing Predcitors and Response

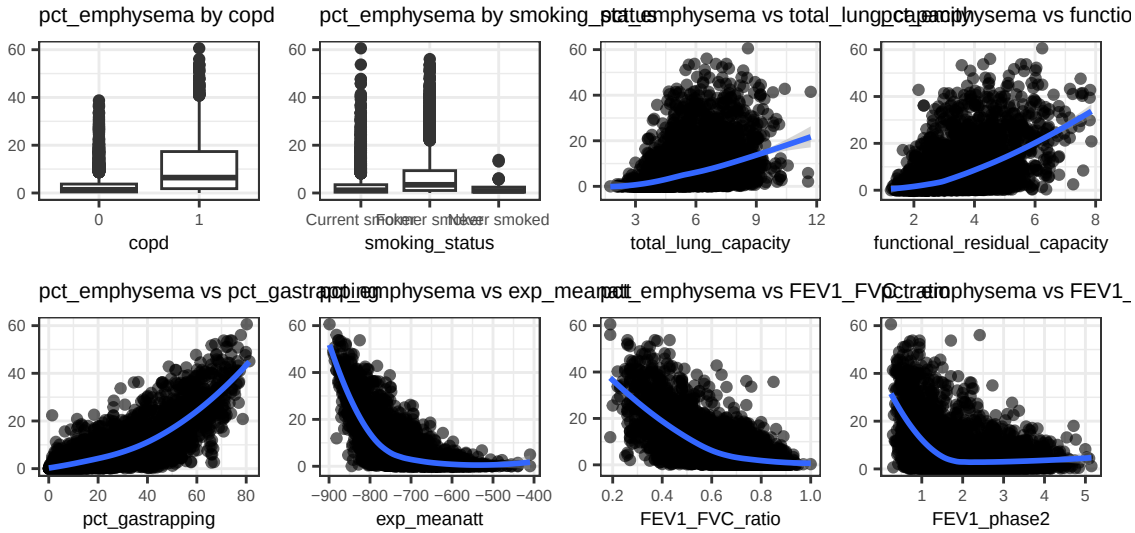
0. Handling Missingness

Before preforming any meaningful statistical analysis, it is vital that we first analyse our missing observations and determine their nature. For a more detailed explanation please consult the appendix. In essence because there is a great proportion of missing variables (specifically the response variable) it has been decided that our analysis will be conducted on two data set: one in which we impute missing the missing dependent variable (**complete**) and one where observations with missing dependent variable with not be taken into account (**imputed**).

i. Transformation of the explanatory variables

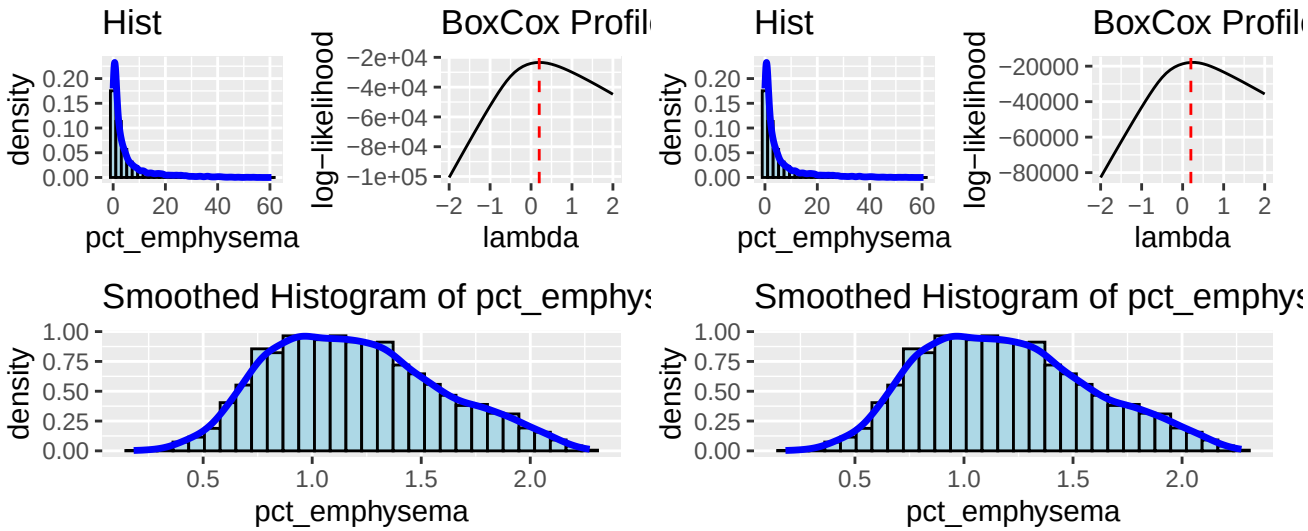
Based on the scatter plots of predictors vs pct_emphysema most relationships seem to be either non existent, linear, quadratic or e^{-x}. At this stage functional relationships cannot be determine but plots give us an idea on what they could be (functional relationships can changed as more variables are included in the model).

Note: The following plots only represent a subset of variables from the data set.



ii. Transformation of the response

The output below consists of three plots: A histogram of the distribution of the response variable (percentage of emphysema), a Box-Cox Power Transformation Plot and a histogram of the distribution of the response variable after applying the optimal box cox transformation. For both **complete** and **imputed** data sets the optimal power transformation is $\frac{1}{5} = 0.2$



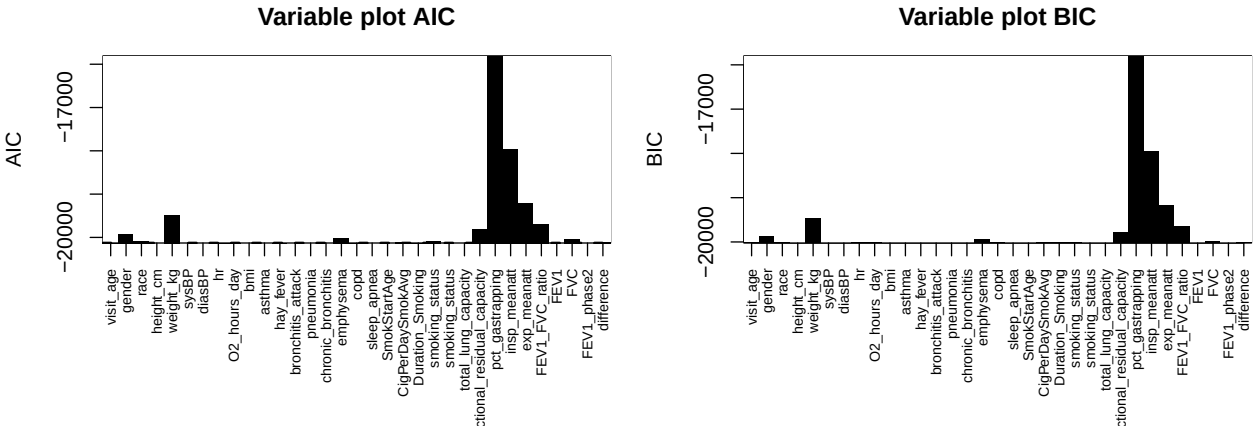
Variable Transformation and Selection

A. Transformation of the explanatory variables: Part 2

In this variable transformation pipeline, we preserve the raw original data for all variables while adding selected transformations for specific lung-function measures. These include square-root and log transforms for lung capacity variables, quadratic terms for mean attenuation metrics, and both logarithmic and shifted negative-exponential transforms for FEV1 in phase 2, enabling flexible modeling while maintaining the original scale for baseline comparisons.
Our best subsets code enforces a mutual exclusivity rule: for variables with multiple transformations (e.g., raw, sqrt, ln), only one version can be chosen per model. This ensures we avoid multicollinearity and select the most useful transformation without redundancy.

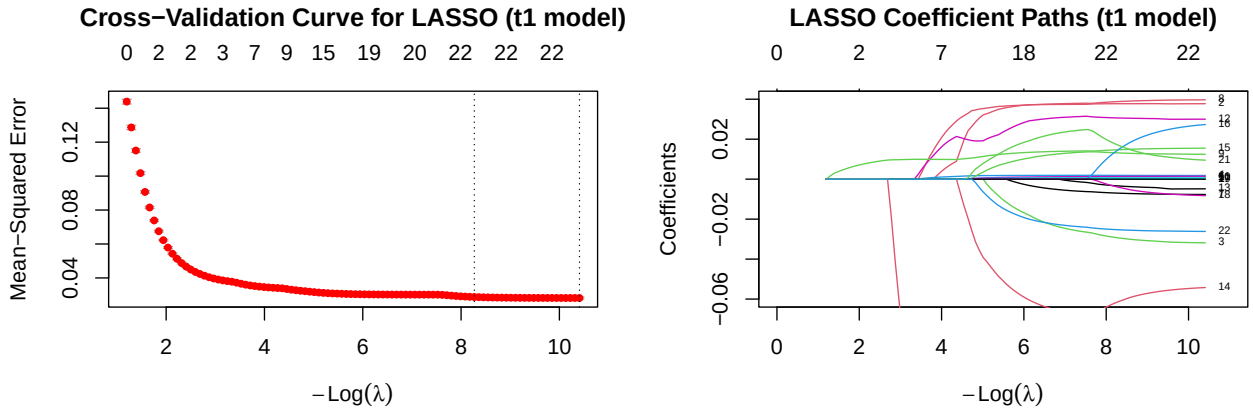
B. Best subset selection + LASSO

Taking into account that our data only contains around 30 informative predictors for regression, utilizing the power of best subset selection is actually feasible. Our results showed that out of the 30 predictors, only 19 of them where meaningful. Yet, at this stage it is still to early to report those as we first need to check whether there exist multicolinearity in our proposed model. So, we ran LASSO on both sets of data **complete** and **imputed**. Both gave the same conclusions, based on our LASSO we dcided to drop 5 variables (visit_age, gender, race, weight_kg, hr)



Given that our ideal choice of lambda (the one that minimizes cross validation error) includes all 19 variables selection from the best subset procedure from previous **code**. We decided to use $5 = -\ln\lambda$ as it is at this point where our coefficients begain to stabilize and our cross validation error does not explode.

Note: Lasso results were consistent across complete and imputed data; plots shown are from complete data.



After preforming LASSO variable selection to reduce multicolinearity, VIF is mostly under control. The only terms the exhibit a high amount of multicolinearity are those quadratic terms.

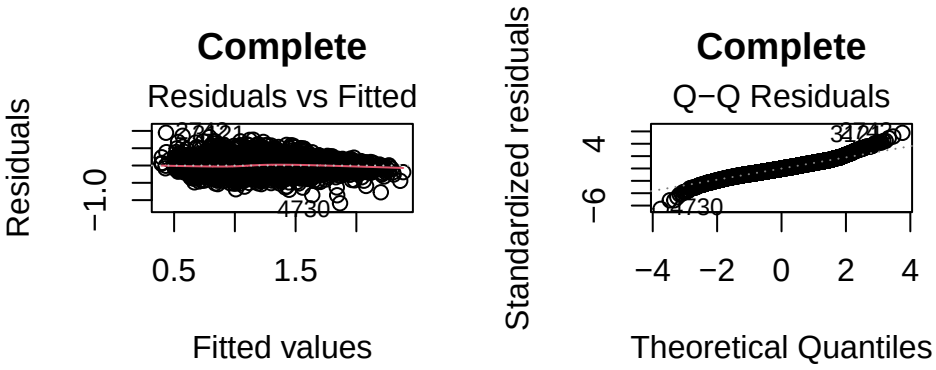
Table 3: VIF Decreasing for both Datasets

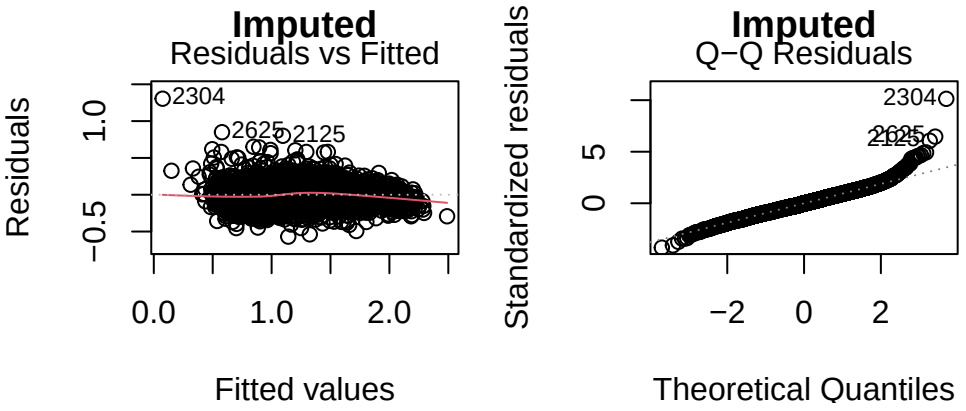
term	GVIF_complete	GVIF_imputed
I(insp_meanatt^2)	867.555	1023.801
insp_meanatt	838.590	992.257
I(exp_meanatt^2)	515.015	884.901
exp_meanatt	427.119	692.826

Statsitical Analysis

A. Residual Analysis

At this point we have a pretty good idea on what variables should be included in our regression model (including their transformations). Yet, before we can draw any inferences we need to make sure that our model meets the 4 major assumptions of Ordinary Least Squares (Zero Mean, Constant Variance, uncorrelated errors and normality of errors).



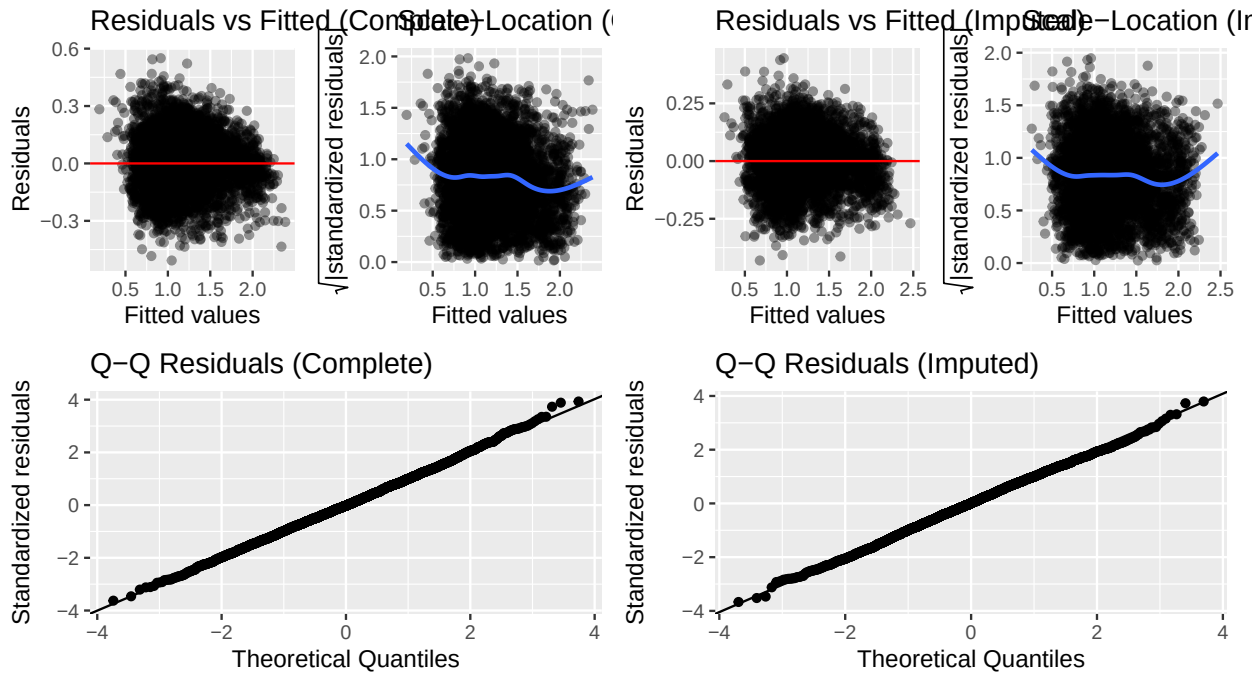


Although the two plots are constructed from different datasets, they point to the same conclusions. The assumption of constant variance is clearly violated. More notably, the normality assumption is strongly violated. In both tails, the residuals deviate substantially from the theoretical quantiles, showing pronounced departures from normality at both the lower and upper extremes. After further investigation we found the culprit.

Both datasets contain imputed values, particularly the **complete** dataset. In many instances we imputed rows with multiple missing entries, sometimes as many as 12. Given the substantial amount of imputation required, it is likely that some of the imputed values were inaccurate, and this may have adversely affected the performance of our linear regression model.

Our further analysis analysiszed outliers for both data sets and determined that in both cases all outliers came from imputed data (erroneous observations where more pronunced in the **complete** dataset). Thus said, we felt that it was appropriate to delete these erroneously computed observations. More detail on how we determined "erroneous" observations in the appendix.

After deleting those problematic residuals, we refit our model and plot our diagnostic plots. As you can observe in both cases our 2 violated assumptions are almost corrected. Our residuals are now normally distributed and our variance is now more consistent across all ranges of fitted values.



B. Model Description, Inference, and Interpretation

i. Model Description

Our final model was chosen through a multi-step procedure. We first identified the predictor set with the lowest AIC, then applied LASSO regularization to remove weak or redundant variables, and finally excluded a small number of observations with erroneous imputations.
Both the complete and imputed datasets produced strong models, with R^2 values of 0.8600341 and 0.9051658, and adjusted R^2 values of 0.8596224 and 0.9048315, respectively. At $\alpha = 0.05$, both models are statistically significant according to their F-tests, indicating that the predictors collectively explain substantial variance in the outcome even though not all individual predictors are significant.

Table 4: Compact Summary of Model Fit Statistics

Model	Residual_SD	MSE	R2	Adj_R2	F_stat	df1	df2	F_pvalue	AIC
Complete	0.140	0.020	0.860	0.860	2089.2	16	5440	0	-5951.8
Imputed	0.118	0.014	0.905	0.905	2707.1	16	4538	0	-6555.6

ii. Model Discription and Interpretation

Let X1 denote hours of supplemental oxygen used per day; X2 indicate hay fever; X3 indicate emphysema; X4 indicate COPD; X5 represent average cigarettes smoked per day; X6 the duration of smoking in years; X7 and X8 the smoking status indicators; X9 the functional residual capacity; X10 the percentage of gas trapping; X11 and X12 the inspiratory mean attenuation and its square; X13 and X14 the expiratory mean attenuation and its square; X15 the FEV1/FVC ratio; and X16 the FVC value. Let Y denote the percentage of emphysema. Then the full model is:

Table 5: Predictor Naming Table (3 Predictors per Column)

x1: O2_hours_day	x2: hay_fever	x3: emphysema
x4: copd	x5: CigPerDaySmokAvg	x6: Duration_Smoking
x7: smoking_status	x8: functional_residual_capacity	x9: pct_gastrapping
x10: insp_meanatt	x11: I(insp_meanatt^2)	x12: exp_meanatt
x13: I(exp_meanatt^2)	x14: FEV1_FVC_ratio	x15: FVC

$$Y = \beta 0 + X1\beta 1 + X2\beta 2 + ... + X16\beta 16 + \varepsilon$$

Our hypotheses for global inference are the following:
H0 : All $\beta_i = 0$
Ha : At least one $\beta_i \neq 0$
for i = 1, 2, ..., 16.

The table summarizes coefficient estimates for 16 variables using both complete-case (C) and imputed (I) analyses, each paired with 95% confidence intervals. Across variables, the imputed estimates generally track closely with the complete-case results, though some values show modest shifts in magnitude or interval range. Overall, the comparison illustrates how imputing missing

data can slightly adjust coefficient estimates while maintaining similar patterns and interpretive conclusions. We did not control for FWER due to the large number of predictors included.

Table 6: Complete-case (C) and imputed (I) coefficient estimates with 95% confidence intervals.

i	$\beta_i^{(C)}$	$L_{95}^{(C)}$	$U_{95}^{(C)}$	$\beta_i^{(I)}$	$L_{95}^{(I)}$	$U_{95}^{(I)}$
1	0.002	0.001	0.003	0.002	0.001	0.003
2	-0.006	-0.014	0.002	-0.008	-0.016	-0.001
3	0.019	0.007	0.032	0.013	0.001	0.024
4	0.015	0.003	0.027	0.014	0.003	0.025
5	0.001	0.001	0.002	0.001	0.001	0.001
6	0.001	0.000	0.001	0.000	0.000	0.001
7	0.019	0.010	0.028	-0.002	-0.011	0.006
8	-0.028	-0.065	0.010	-0.040	-0.073	-0.007
9	-0.032	-0.039	-0.024	-0.011	-0.019	-0.003
10	0.022	0.021	0.023	0.031	0.030	0.031
11	0.037	0.034	0.040	0.044	0.041	0.048
12	0.000	0.000	0.000	0.000	0.000	0.000
13	-0.017	-0.019	-0.016	-0.029	-0.031	-0.028
14	0.000	0.000	0.000	0.000	0.000	0.000
15	-0.311	-0.361	-0.262	-0.293	-0.340	-0.246
16	0.015	0.009	0.021	-0.004	-0.010	0.002

Conclusion

In summary, our analysis identifies a clear set of health related variables that play the strongest role in predicting the percentage of emphysema present in a patient's lungs. Measures tied directly to respiratory function, such as lung capacity metrics, gas trapping indicators, and attenuation based imaging variables, emerged as the most influential predictors. This aligns with expectations, as these metrics reflect the structural and functional deterioration of lung tissue. Smoking behavior, including long term exposure and daily quantity, also contributed meaningfully to the prediction of emphysema percentage. While smoking is widely recognized as a primary cause of emphysema, our regression results confirm that smoking related variables are consistently associated with higher emphysema severity. In contrast, several general health indicators such as BMI, heart rate, and blood pressure were not as impactful as originally hypothesized, suggesting that emphysema severity is driven more by respiratory specific factors than by broader measures of overall health.
In essence, both data set more or less came to the same conclusions with slight variations is coefficients and their respective CI's. Overall, it seems like the only visible effect imputation had was producing more outliers, drastic enough to perturbate results but not drastic enough to where results from both datasets where drastically different after account for outleirs.
As with any statistical study, limitations must be acknowledged. The dataset reflects health measurements at a single point in time, even though emphysema develops progressively. Like mentioned in the introduction linear regression also assumes additive relationships between predictors, which may oversimplify the complex interactions underlying lung deterioration. Additionally, the need to impute missing values in both datasets reduced data quality and led to the deletion of several observations, ultimately making the model less powerful. Nevertheless, linear regression remains valuable in this context because it offers clear interpretability, making it well suited for identifying which variables contribute most strongly to emphysema percentage. While this study confirms linear regression's utility as a transparent and methodologically sound first step in emphysema research, it also highlights opportunities for methodological advancement. Future investigations should in-

corporate longitudinal data to model temporal dynamics and employ nonlinear approaches (e.g., generalized additive models, machine learning techniques) to capture the complex interactions inherent in respiratory pathophysiology. These methodological expansions would complement-not invalidate-the foundational relationships identified through linear regression, creating a more comprehensive understanding of emphysema progression.

Refrences

1. Mayo Clinic Staff. Emphysema: Symptoms and Causes. Mayo Clinic. https://www.mayoclinic.org/diseases-conditions/emphysema/symptoms-causes/syc-20355555.
2. Cleveland Clinic Staff. Emphysema: Causes, Symptoms, and Treatment. Cleveland Clinic. https://my.clevelandclinic.org/health/diseases/9370-emphysema.
3. Hasenstab, K. Lecture Slides on Linear Models, Regression Diagnostics, and Variable Selection. Department of Statistics, [San Diego State University], 2025.
4. OpenAI. (2025). *ChatGPT* [Large language model]. https://chat.openai.com/

Appendix

Data Preprocess

We will start our appendix by describing how our data was preprocessed.
We begin by loading the COPD dataset and cleaning sentinel values: continuous variables with -1 are recoded to NA, and categorical values labeled "unknown" (and "missing" for sleep_apnea) are also set to NA. Next, several Yes/No respiratory diagnoses are converted to 0/1 indicators and then coerced to factors, along with demographic and smoking-status variables, so they can be treated as categorical predictors. We then relocate pct_emphysema to appear directly after sid for convenience. Finally, for individuals who report "Never smoked," we set SmokStartAge, CigPerDaySmokAvg, and Duration_Smoking to 0, reflecting the absence of smoking history rather than missing data.

```
# Load Data ------------------------------------------------------------

# Read the COPD dataset from a CSV file into a data frame
df_copd = read_csv( "/cloud/project/copd_data.csv" )


# Replace -1's with NA's -----------------------------------------------

# For various continuous / numeric variables, recode negative sentinel values
# (e.g., -1) to NA to represent missingness.
df_copd = df_copd |>
  mutate(
    # Systolic blood pressure: set negative values to NA
    sysBP = ifelse( sysBP < 0, NA, sysBP ),
    # Diastolic blood pressure: set negative values to NA
    diasBP = ifelse( diasBP < 0, NA, diasBP ),
    # Heart rate: set negative values to NA
    hr = ifelse( hr < 0, NA, hr ),
    # Recode hay_fever numeric codes into categorical labels
    hay_fever = case_when(
      hay_fever == 0 ~ "No",
      hay_fever == 1 ~ "Yes",
      hay_fever == 3 ~ "unknown"
    ),
    # Smoking-related variables: convert -1 sentinel to NA
    SmokStartAge = ifelse( SmokStartAge == -1 , NA, SmokStartAge),
    CigPerDaySmokAvg = ifelse( CigPerDaySmokAvg == -1, NA, CigPerDaySmokAvg),
    Duration_Smoking = ifelse( Duration_Smoking == -1, NA, Duration_Smoking),
    # Lung function / imaging measures: convert -1 sentinel to NA
    total_lung_capacity = ifelse( total_lung_capacity == -1 , NA, total_lung_capacity),
    pct_emphysema = ifelse( pct_emphysema == -1 , NA, pct_emphysema),
    functional_residual_capacity = ifelse( functional_residual_capacity == -1 , NA, functional
    pct_gastrapping = ifelse( pct_gastrapping == -1 , NA, pct_gastrapping),
    insp_meanatt = ifelse( insp_meanatt == -1, NA, insp_meanatt),
    exp_meanatt = ifelse( exp_meanatt == -1, NA, exp_meanatt),
```

```
    FEV1_FVC_ratio = ifelse( FEV1_FVC_ratio == -1, NA, FEV1_FVC_ratio),
    FEV1 = ifelse( FEV1 == -1, NA, FEV1),
    FVC = ifelse( FVC == -1, NA, FVC)
  )


# Convert categorical "unknown" to NA's ----------------------------------

# For categorical variables, convert the "unknown" (and "missing" for sleep_apnea)
# categories into NA to treat them as missing.
df_copd = df_copd |> mutate(
  asthma = ifelse( asthma == "unknown", NA, asthma),
  hay_fever = ifelse( hay_fever == "unknown", NA, hay_fever),
  bronchitis_attack = ifelse( bronchitis_attack == "unknown", NA, bronchitis_attack),
  pneumonia = ifelse( pneumonia == "unknown", NA, pneumonia),
  chronic_bronchitis = ifelse( chronic_bronchitis == "unknown", NA, chronic_bronchitis),
  emphysema = ifelse( emphysema == "unknown", NA, emphysema),
  copd = ifelse( copd == "unknown", NA, copd),
  # Sleep apnea: treat both "unknown" and "missing" as NA
  sleep_apnea = ifelse( (sleep_apnea == "unknown") | (sleep_apnea == "missing") , NA, sleep_apn
)


# Convert categorical to binary ------------------------------------------

# Convert Yes/No categorical variables into numeric 0/1 indicators,
# preserving NA where values are missing.
df_copd <- df_copd |>
  mutate(
    asthma = case_when(
      is.na(asthma) ~ NA_real_,  # keep missing as NA
      asthma == "No" ~ 0,
      asthma == "Yes" ~ 1
    ),
    hay_fever = case_when(
      is.na(hay_fever) ~ NA_real_,
      hay_fever == "No" ~ 0,
      hay_fever == "Yes" ~ 1
    ),
    bronchitis_attack = case_when(
      is.na(bronchitis_attack) ~ NA_real_,
      bronchitis_attack == "No" ~ 0,
      bronchitis_attack == "Yes" ~ 1
    ),
    pneumonia = case_when(
      is.na(pneumonia) ~ NA_real_,
      pneumonia == "No" ~ 0,
```

```
      pneumonia == "Yes" ~ 1
    ),
    chronic_bronchitis = case_when(
      is.na(chronic_bronchitis) ~ NA_real_,
      chronic_bronchitis == "No" ~ 0,
      chronic_bronchitis == "Yes" ~ 1
    ),
    emphysema = case_when(
      is.na(emphysema) ~ NA_real_,
      emphysema == "No" ~ 0,
      emphysema == "Yes" ~ 1
    ),
    copd = case_when(
      is.na(copd) ~ NA_real_,
      copd == "No" ~ 0,
      copd == "Yes" ~ 1
    ),
    sleep_apnea = case_when(
      is.na(sleep_apnea) ~ NA_real_,
      sleep_apnea == "No" ~ 0,
      sleep_apnea == "Yes" ~ 1
    )
  )


# Convert into factors ------------------------------------------------

# Coerce selected variables to factors for categorical analysis/models.
df_copd = df_copd |>
  mutate(
    gender = as.factor(gender),
    race = as.factor(race),
    asthma = as.factor(asthma),
    hay_fever = as.factor(hay_fever),
    bronchitis_attack = as.factor(bronchitis_attack),
    pneumonia = as.factor(pneumonia),
    chronic_bronchitis = as.factor(chronic_bronchitis),
    emphysema = as.factor(emphysema),
    copd = as.factor(copd),
    sleep_apnea = as.factor(sleep_apnea),
    smoking_status = as.factor(smoking_status),
    gender = as.factor(gender)  # redundant re-coercion of gender to factor
  )

# Relocate percent of emphysema ---------------------------------------

# Move pct_emphysema column to immediately follow sid in the data frame.
```

```r
df_copd = df_copd |>
  relocate( pct_emphysema, .after = sid )

# Zeros for smoke age and CigPerDaySmokAvg -------------------------------

# For non-smokers ("Never smoked"), set smoking-related measures to 0
# instead of leaving them as NA or original values.
df_copd <- df_copd |>
  mutate(
    SmokStartAge = if_else(smoking_status == "Never smoked", 0, SmokStartAge),
    CigPerDaySmokAvg = if_else(smoking_status == "Never smoked", 0, CigPerDaySmokAvg),
    Duration_Smoking = if_else(smoking_status == "Never smoked", 0, Duration_Smoking)
  )
```

The subsequent sections of the appendix provide supplementary material, organized to mirror the structure of the main analysis.

Explanation of variables used

Below are the original 35 variables as well as their description (not all where used or relevant in the analysis):

- sid: The anonymized patient identification number.
- visit_year: The calendar year in which the patient visit occurred.
- visit_date: The specific date on which the patient visit occurred.
- visit_age: The patient's age at the time of the visit.
- gender: The patient's reported gender (Male or Female).
- race: The patient's race category (White, Black or African American).
- height_cm: The patient's height measured in centimeters.
- weight_kg: The patient's weight measured in kilograms.
- sysBP, diasBP: Systolic and diastolic blood pressure, respectively.
- hr: The patient's heart rate.
- O2_hours_day: For a typical 24-hour day, the number of hours of supplemental oxygen used.
- bmi: The patient's body mass index.
- asthma: Whether the patient has ever been diagnosed with asthma (Yes, No).
- hay_fever: Whether the patient has ever had hay fever (Yes, No).
- bronchitis_attack: Whether the patient has ever had a bronchitis attack (Yes, No).
- pneumonia: Whether the patient has ever had pneumonia (Yes, No).
- chronic_bronchitis: Whether the patient has ever been diagnosed with chronic bronchitis (Yes, No).
- emphysema: Whether the patient has ever been diagnosed with emphysema (Yes, No).
- copd: Whether the patient has been diagnosed with chronic obstructive pulmonary disease (Yes, No).
- sleep_apnea: Whether the patient has ever had sleep apnea (Yes, No).
- SmokStartAge: The age at which the patient began cigarette smoking.
- CigPerDaySmokAvg: The average number of cigarettes smoked per day across smoking history.
- Duration_Smoking: The number of years the patient has smoked.
- smoking_status: Categorical indicator of smoking behavior (Never smoked, Former smoker, Current smoker).
- total_lung_capacity: The lung volume at full inspiration, measured in liters.
- pct_emphysema: The percentage of emphysematous (damaged) lung tissue.
- functional_residual_capacity: The volume of air remaining in the lungs at the end of expiration, in liters.
- pct_gastrapping: The percentage of air trapping present after exhalation.
- insp_meanatt: The average lung density at full inspiration, measured in Hounsfield units.
- exp_meanatt: The average lung density at expiration, measured in Hounsfield units.
- FEV1_FVC_ratio: The ratio between forced expiratory volume in 1 second (FEV1) and forced vital capacity (FVC).
- FEV1: The volume of air forcefully exhaled in 1 second.
- FVC: The total exhaled air volume after a full inhalation.
- FEV1_phase2: The FEV1 value measured five years later during a follow-up assessment.

Full table summaries and their distributions

Originally, due to the lack of space in the study and lack of relevance of some variables most variables where excluded from our table summaries and their distributions where not displayed.

Table 7: Table continues below

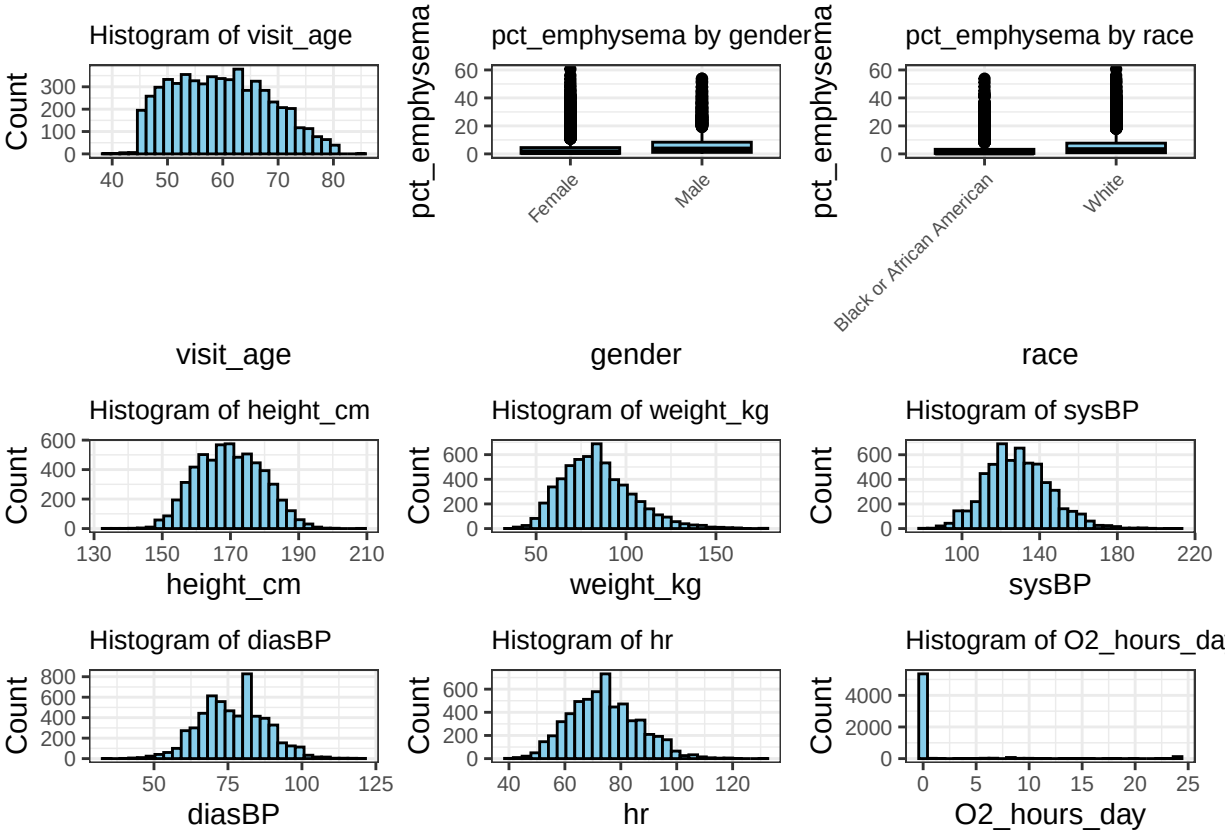
variable	unique	missing	mean	sd	min	median	max
sid	5747	0	NA	NA	NA	NA	NA
pct_emphysema	4698	1045	5.582	8.429	0.0001963	2.108	60.58
visit_year	4	0	2009	0.8079	2008		
visit_date	40	0	NA	NA	NA		
visit_age	392	0	59.75	8.688	39		
gender	2	0	NA	NA	NA		
race	2	0	NA	NA	NA		
height_cm	437	0	169.9	9.534	133.7		
weight_kg	833	0	84.14	19.59	34.9		
sysBP	111	2	128.8	16.58	80		
diasBP	76	2	76.83	10.75	34		
hr	81	1	74.17	12.18	40		
O2_hours_day	23	0	0.9468	4.095	0		
bmi	2052	0	29.08	6.141	12.67		
asthma	2	380	NA	NA	NA		
hay_fever	2	435	NA	NA	NA		
bronchitis_attack	2	485	NA	NA	NA		
pneumonia	2	283	NA	NA	NA		
chronic_bronchitis	2	450	NA	NA	NA		
emphysema	2	410	NA	NA	NA		
copd	2	400	NA	NA	NA		
sleep_apnea	2	602	NA	NA	NA		
SmokStartAge	45	0	16.74	4.821	0		
CigPerDaySmokAvg	47	0	23.78	11.42	0		
Duration_Smoking	443	3	35.05	10.8	0		
smoking_status	3	0	NA	NA	NA		
total_lung_capacity	4462	1045	5.545	1.407	1.746		
functional_residual_capacity	3810	1650	3.18	1.006	1.242		
pct_gastrapping	4088	1650	19.81	17.72	0.04412		
insp_meanatt	4617	1045	-830.8	36.53	-925.1		
exp_meanatt	4059	1650	-707.5	70.11	-898.8		
FEV1_FVC_ratio	78	29	0.6882	0.1418	0.19		
FEV1	2634	29	2.337	0.8523	0.283		
FVC	2820	29	3.369	0.9749	0.593		
FEV1_phase2	2628	0	2.127	0.8382	0.246		

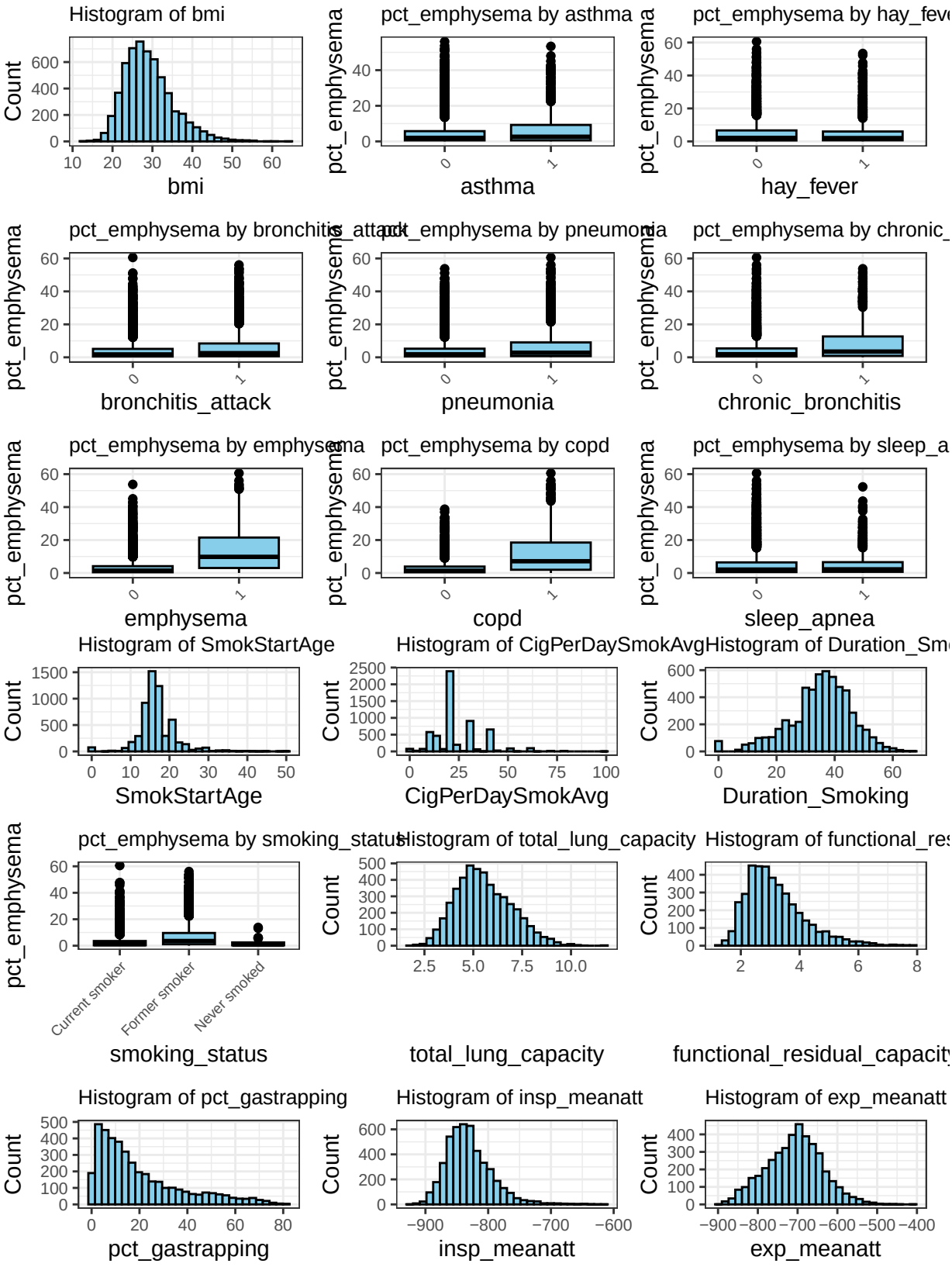
median	max
2009	2011
NA	NA
59.5	85
NA	NA
NA	NA
170	208.3
82	176.4
128	211
77	120
73	131
0	24
28.2	64.1
NA	NA
NA	NA
NA	NA
NA	NA
NA	NA
NA	NA
NA	NA
NA	NA
16	50
20	99
36	67
NA	NA
5.395	11.7
2.982	7.861
13.63	81.27
-836	-615
-703.7	-409.7
0.73	1
2.311	5.26
3.266	7.106
2.101	5.147

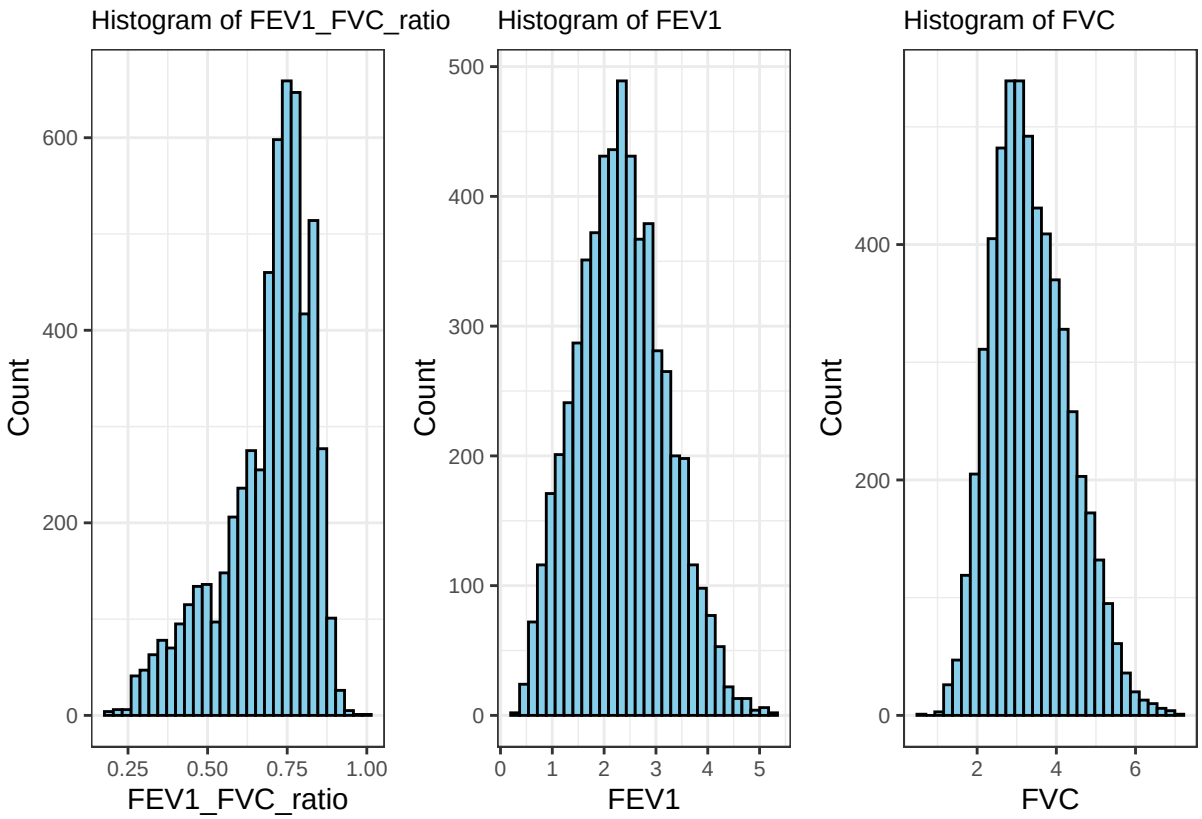
Histograms and Boxplots

To provide readers with a clear understanding of the underlying data and to verify the absence of irregularities, we include this section of histograms and boxplots. These visualizations allow us to inspect the distributions of key predictors and examine the relationship between categorical variables and pct_emphysema. Overall, the plots indicate that the data behave as expected, with no evident anomalies or problematic patterns, giving us confidence in the quality and suitability of the dataset for subsequent analyses.

Variables such as weight, height, and heart rate appear to be approximately normally distributed, as expected for general physiological measures. In contrast, most respiratory disease variables do not show clear evidence of normality. However, COPD and emphysema stand out as exceptions, displaying distributions that differ noticeably from the others.




Histogram of bmi
Count
bmi
pct_emphysema by asthma
pct_emphysema
asthma
pct_emphysema by hay_fev
hay_fever
pct_emphysema by bronchitis_attack
bronchitis_attack
pct_emphysema by pneumonia
pneumonia
pct_emphysema by chronic_
chronic_bronchitis
pct_emphysema by emphysema
emphysema
pct_emphysema by copd
copd
pct_emphysema by sleep_a
sleep_apnea
Histogram of SmokStartAge
SmokStartAge
Histogram of CigPerDaySmokAvg
CigPerDaySmokAvg
Histogram of Duration_Sm
Duration_Smoking
pct_emphysema by smoking_status
Current smoker
Former smoker
Never smoked
smoking_status
Histogram of total_lung_capacity
total_lung_capacity
Histogram of functional_res
functional_residual_capacity
Histogram of pct_gastrapping
pct_gastrapping
Histogram of insp_meanatt
insp_meanatt
Histogram of exp_meanatt
exp_meanatt


Histogram of FEV1_FVC_ratio
Count
FEV1_FVC_ratio
0
200
400
600
0.25
0.50
0.75
1.00
Histogram of FEV1
Count
FEV1
0
100
200
300
400
500
0
1
2
3
4
5
Histogram of FVC
Count
FVC
0
200
400
2
4
6

A. Cleaning our Data

The preprocessing steps clean and standardize the COPD dataset by replacing all negative numeric values with missing values (NA) and recoding several categorical variables. Unknown or missing categories in health-related variables (e.g., asthma, hay fever, COPD, sleep apnea) are converted to NA, and then all Yes/No variables are transformed into binary indicators before being stored as factors. Additional factor conversions are applied to demographic and smoking variables. The dataset is reorganized by moving pct_emphysema next to the subject ID, and a final COPD dataset is created by removing identifier and visit information, leaving only relevant analytic variables.

Dealing with Missing Data

Before preforming any meaningful statistical analysis, it is vital that we first analyse our missing observations and determine their nature.
For starters, we should note that the following variables have less then 3 missing observations: Systolic blood pressure, Diastolic blood pressure and Heart rate. This minimal missingness is unlikely to meaningfully impact statistical inferences or bias parameter estimates.

Missing Respiratory Disease predictors

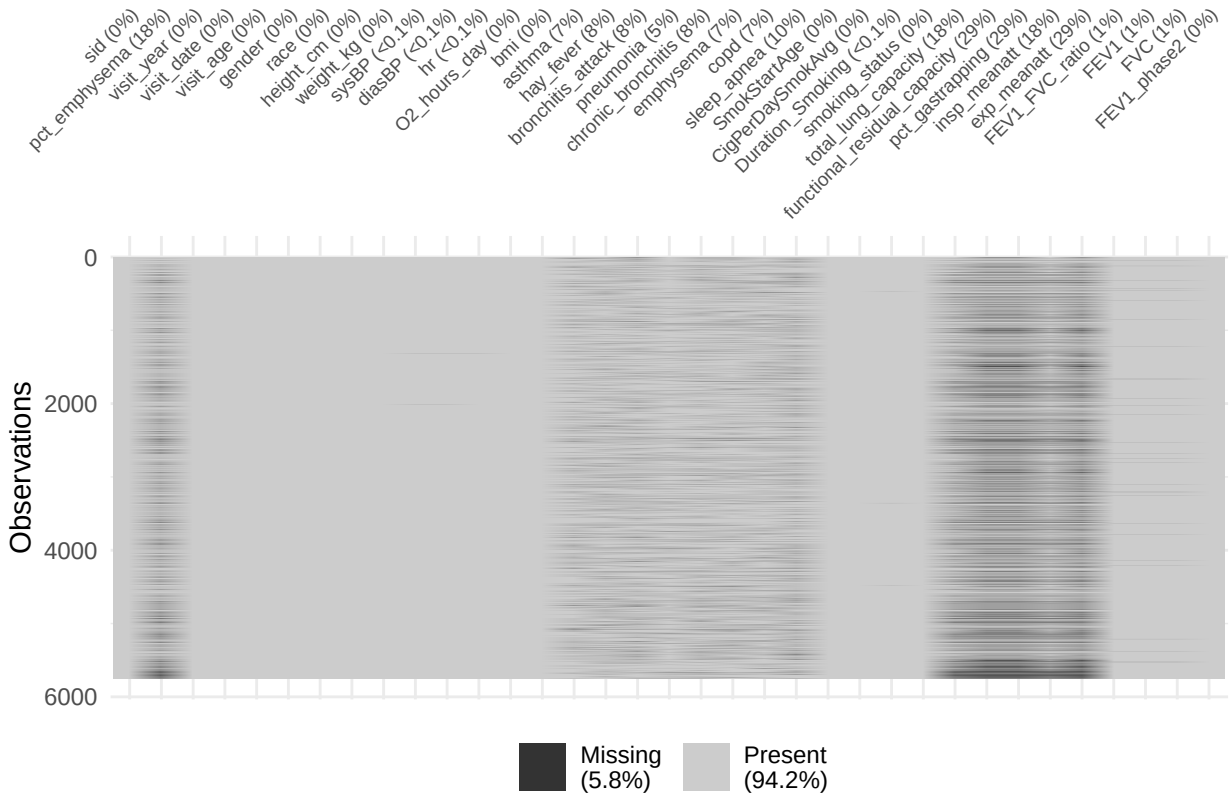
In previous analysis, it seemed like the predictors the **explained/indicated(change)** whether said patient had that indicated respiratory disease has missing values. However, this is not entirely accurate as in reality those observations marked NA are actually **unknown**. It was ultimately decided to convert those Unknown to NA's and then impute those NA's. This 'unknown' category was not included in the model because, for most variables, it provides little predictive information and would not meaningfully change the results.

SmokStartAge, CigPerDaySmokAvg, Duration_Smoking

At first glance: SmokStartAge, CigPerDaySmokAvg, Duration_Smoking appear to be missing as 76 observation are labeled as -1, which is impossible based on what our predictors are explaining. However, after further investigation it should be noted that -1 refers to not applicable. If we observed the smoking status variable, we can see that the missingness is explained by the fact that these 76 individuals have never smoked, thus SmokStartAge, CigPerDaySmokAvg, Duration_Smoking does not apply to them. Instead of labeling these observations as NA, it has been decided that instead they will be labeled as 0.

Other missing variables

It should be noted that after further analysis, the rest of the missing data appear to be missing **Completely at random**. Although dataseems to be **missing completely at random**, serval variable are missing together. pct_emphysema, total_lung_capaci and insp_meanatt are missing together likely as a result of tests not being conducted due to either logistical constraints (e.g., unavailability of necessary equipment or specialized personnel) or participant-related factors (unwillingness or inability to undergo the procedure) (likely used the same test to calculate those three variables). The same applies for functional_residual_capacity, exp_meanatt and pct_gastrapping as they are always missing together. Lastly, with only 29 missing observations FEV1_FVC_ratio, FEV1 and FVC are also missing together either due to unwillingness or inability to undergo the procedure (this applied to FEV1 and FVC only, FEV1_FVC_ratio is a ratio of the previous mentioned variables and as a result is missing when the previous variables are also missing).



Missing response variable

It should be noted that our response variable Percentage of emphysema (damaged lung areas) denoted as pct_emphysema has 1045 missing observations (20% of response observations are missing). Unfortunately, dealing with missing response variable is harder then dealing with explanatory variables as the tampering with this response variable will likely have bigger consequences in our analysis of data. Although opinions differ, most sources would tell us that imputing 20% of our response variable is dangerous, therefore our further statistical analysis will be conducted twice: once on a dataset that imputes missing response and once where those observations with missing response variables will not be included.

Note 1: Data was imputed using the mice package where method = "rf" (Random forest imputations).

Note 2: In the original code **complete** and **imputed** dataset where downloaded from project files. They where imputed with the same method below, however imuting such large datasets took a long time and it just wan't viable imputing each time we wanted to run our R code.

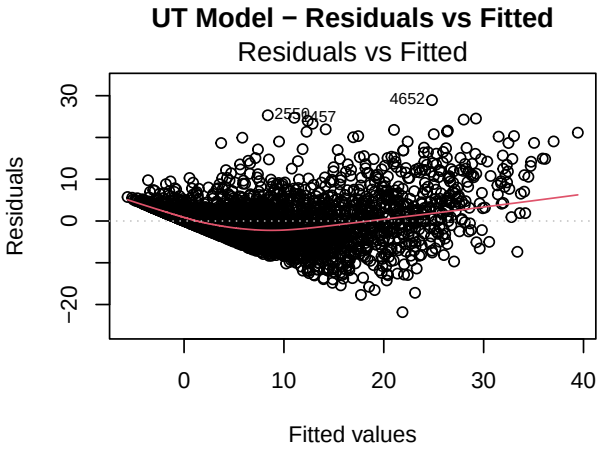
```
# ---- Helper: build methods vector based on missingness ----
prepare_methods <- function(df, methods_vector, yvar = "pct_emphysema") {
  mvec <- methods_vector

  # Do NOT impute Y if it has no missing data
  if (all(!is.na(df[[yvar]]))) {
    mvec[yvar] <- ""
  }
```

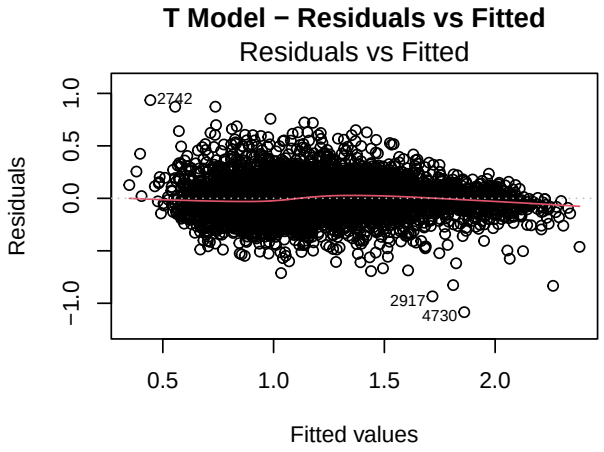
```r
  # Do NOT impute variables with no missingness
  for (v in names(mvec)) {
    if (all(!is.na(df[[v]]))) {
      mvec[v] <- ""
    }
  }

  return(mvec)
}

# Refill cigarites with 0's
copd = copd |>
  mutate( SmokStartAge = ifelse( is.na(SmokStartAge), 0 , SmokStartAge),
          CigPerDaySmokAvg = ifelse( is.na(CigPerDaySmokAvg), 0 , CigPerDaySmokAvg),
          Duration_Smoking = ifelse( is.na(Duration_Smoking), 0 , Duration_Smoking) )

# Observations 468, 3351 are truely missing and not just 0
copd[ 486 ,]$"Duration_Smoking" = NA
copd[ 3351 ,]$"Duration_Smoking" = NA

# Three dataframes: Complete , Imputed and non-missing
complete = copd
imputed = copd[ !is.na( copd$pct_emphysema ) , ]
non_missing = na.omit(copd)

# Preprocess data data. transform pct_emphizyme to logistic
complete = complete |>
  mutate( pct_emphysema = pct_emphysema * (1/100) ) |>
  mutate( pct_emphysema = log( (pct_emphysema)/(1-pct_emphysema) ) )


# ============================================================
# 1. COMPLETE → Impute Y and all other missing values
# ============================================================
mice_complete <- mice(
  complete,
  method = "rf",
  m = 6,
  maxit = 10,
  seed = 123
)

# OVERWRITE original complete data frame
complete <- complete(mice_complete)

# ============================================================
# 2. IMPUTED → Y already complete → only impute other vars
```

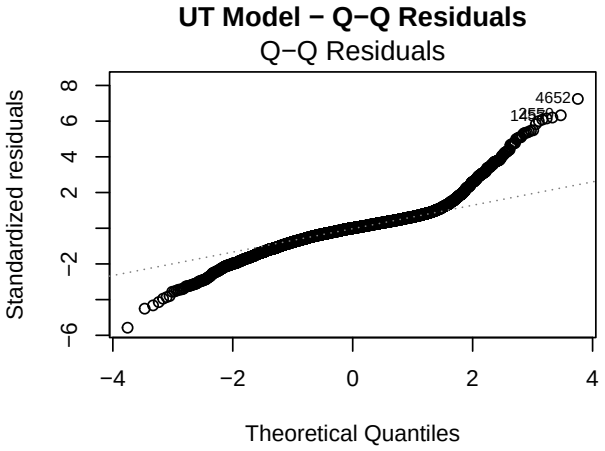
```r
# ============================================================
methods_imputed <- prepare_methods(imputed, methods_vector)

mice_imputed <- mice(
  imputed,
  method = "rf",
  m = 6,
  maxit = 10,
  seed = 123
)

# OVERWRITE original imputed data frame
imputed <- complete(mice_imputed)
```

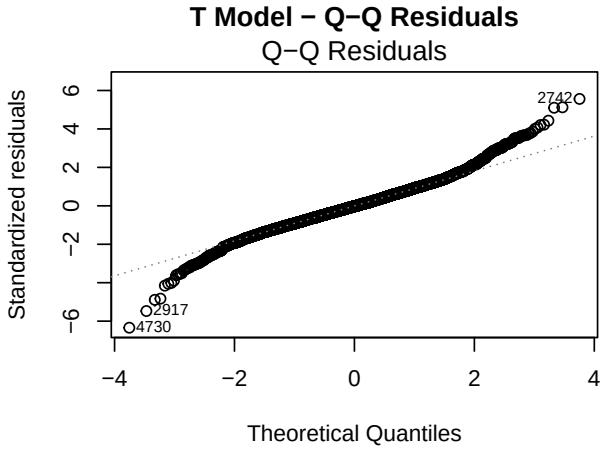
Boxcox justification

Although the optimal power transformation may seem arbitrary (in many cases it is), it is a useful tool that allows us to stabilize our model in order to conduct inference. In the plots below are a comparison between residual plots and normality plots for transformed and transformed data.
In the plots below we can observed how the untransformed data **UT** presents more severe violations in our 4 OLS assumptions.
Note: Normality looks fuzzy as outliers are still present.









Deletion of Outliers Due to Erroneous Imputation

Below is an in-depth, line-by-line explanation of what your code is doing. Conceptually, both code chunks do the same type of work:

1. Compute standard influence diagnostics from a fitted linear model.
2. Combine them into a data frame.
3. Define "rules" (thresholds) for what counts as an outlier/influential observation.
4. Count how many rules each observation violates.
5. Select "bad" observations with many violations.
6. Optionally check how many of those "bad" observations correspond to imputed data.
7. Refit the model after removing those influential rows.

I'll first walk through the "complete" code block, then highlight the parallel steps for the "imputed" block and note the differences.

We start by extracting the core influence diagnostics from the fitted regression model(leverage, studentized deleted residuals, cook's distance and dfbetas) so that each observation's impact on the model can be systematically evaluated. The hatvalues() function computes each observation's leverage, which reflects how unusual its predictor values are relative to the rest of the data set. dffits() measures how much an observation changes its own fitted value when removed, while cooks.distance() summarizes how much the entire set of fitted values would shift if that observation were excluded. dfbetas() gives a coefficient-specific measure, showing how much each regression coefficient would change when the observation is left out. Together, these diagnostics capture different aspects of influence: unusual predictor patterns, local and global effects on fitted values, and instability in estimated coefficients. The final two lines compute the number of observations (n) and the number of model parameters (p), which are needed to set rule-of-thumb thresholds for identifying influential points. This step forms the foundation for all subsequent influence assessment.
Note: prompt said to assumed reader is non-statistician, therfore extra details are added.

```
# 1. Extract diagnostics ----------------------------------------------------

complete_lev     <- hatvalues(complete_model_t2)
complete_dffits  <- dffits(complete_model_t2)
complete_cooks   <- cooks.distance(complete_model_t2)
complete_dfbetas <- dfbetas(complete_model_t2)

complete_n <- length(complete_lev)
complete_p <- length(coef(complete_model_t2))

# 2. Influence dataframe ----------------------------------------------------

complete_influence_df <- data.frame(
  Observation   = seq_len(complete_n),
  Leverage      = complete_lev,
  DFFITS        = complete_dffits,
  CooksDistance = complete_cooks,
  complete_dfbetas    # one column per coefficient
)
```

```r
# 3. Thresholds ------------------------------------------------------------

complete_lev_threshold     <- 2 * complete_p / complete_n
complete_dffits_threshold  <- 2 * sqrt(complete_p / complete_n)
complete_cooks_threshold   <- 4 / complete_n
complete_dfbetas_threshold <- 2 / sqrt(complete_n)
```

Explaining residual "rule"

The code applies a set of standard regression-diagnostic "rules" to determine whether an observation is unusually influential. These rules evaluate different ways a data point can distort the fitted model. Leverage is compared against the rule-of-thumb cutoff $2p/n$ to identify observations whose predictor patterns are far from the bulk of the data. DFFITS is checked against $2\sqrt{(p/n)}$ to see whether removing an observation would significantly change its own fitted value, while Cook's distance, compared to $4/n$, detects observations that substantially alter the model's fitted values overall. For each coefficient, DFBETAS are also examined using the threshold $2/\sqrt{n}$, revealing points that exert excessive influence on specific regression parameters. Together, these rules capture multiple dimensions of influence: unusual predictor values, local and global impact on fitted values, and instability in individual coefficients. Because the dataset contains many predictors, many observations, and substantial missingness, it becomes significantly harder to examine residuals and influence patterns by visual inspection alone. These automated rules and aggregated violation scores therefore provide a structured and scalable way to identify influential observations in a complex modeling setting.

```r
# 4. DFBETAS violation counts ----------------------------------------------

complete_dfbetas_violation_count <- apply(
  abs(complete_dfbetas) > complete_dfbetas_threshold,
  1,
  sum
)

# 5. Total violation score -------------------------------------------------

complete_total_violation_score <-
  (complete_lev > complete_lev_threshold) +
  (abs(complete_dffits) > complete_dffits_threshold) +
  (complete_cooks > complete_cooks_threshold) +
  complete_dfbetas_violation_count

# 6. Final influence dataframe ---------------------------------------------

complete_final_df <- cbind(
  complete_influence_df,
  DFBETAS_Violation_Count = complete_dfbetas_violation_count,
  TotalViolations         = complete_total_violation_score
)
```

```r
# 7. Identify violators ------------------------------------------------

complete_violators_df <- subset(complete_final_df, TotalViolations > 0)

# Non-leverage violators (any of the non-leverage rules or DFBETAS) -------
complete_violators_non_leverage <- complete_final_df %>%
  filter(
    abs(DFFITS) > complete_dffits_threshold |
      CooksDistance > complete_cooks_threshold |
      DFBETAS_Violation_Count > 0
  ) %>%
  arrange(desc(TotalViolations))
```

In this step, you identify which observations are influential enough to be removed from the model based on the diagnostic rules you previously calculated. You start with the set of observations that violated at least one *non-leverage* rule (DFFITS, Cook's distance, or DFBETAS). From this subset, you then apply a strict threshold-here, requiring a TotalViolations score of 8 or more-to isolate only the most problematic cases. The `filter(TotalViolations >= 8)` command keeps only those observations that break many influence rules simultaneously, indicating they have a disproportionately large impact on the model. Finally, `pull(Observation)` extracts the row numbers of these influential data points, producing `complete_remove_idx`, a vector of indices that will be removed before refitting the regression model. This step formalizes the decision about which observations are too influential to keep in the analysis.

```r
# 8. Row indices to remove for the complete model -------------------------

# (adjust threshold here if you want, e.g. >= 8)
complete_remove_idx <- complete_violators_non_leverage %>%
  filter(TotalViolations >= 8) %>%
  pull(Observation)
```

```r
# 9. Percentage of removed rows that come from imputed data ---------------

complete_is_imputed <- complete_remove_idx %in% which(!complete.cases(df_copd))
percent_violation_complete <- mean(complete_is_imputed)
# percent_violation_complete is between 0 and 1

# 10. Refit model without influential observations ------------------------

complete_model_t3 <- complete[-complete_remove_idx, ] %>%
  lm(
    pct_emphysema ~
      O2_hours_day +
      hay_fever +
      emphysema +
      copd +
      CigPerDaySmokAvg +
      Duration_Smoking +
```

```
      smoking_status +
      functional_residual_capacity +
      pct_gastrapping +
      insp_meanatt + I(insp_meanatt^2) +
      exp_meanatt + I(exp_meanatt^2) +
      FEV1_FVC_ratio +
      FVC,
    data = .
  )
```

The same logic and rules apply to the **imputed** data set ass well. A small disclaimer here is that after the deletion of residuals observation 4071 and 4072 where manually deleted as they where influential enough to influence our regression line but were not capture by the mechanism described above.

```
options(scipen = 999)

# 1. Extract diagnostics ----------------------------------------------------

imputed_lev     <- hatvalues(imputed_model_t2)
imputed_dffits  <- dffits(imputed_model_t2)
imputed_cooks   <- cooks.distance(imputed_model_t2)
imputed_dfbetas <- dfbetas(imputed_model_t2)

imputed_n <- length(imputed_lev)
imputed_p <- length(coef(imputed_model_t2))

# 2. Influence dataframe ----------------------------------------------------

imputed_influence_df <- data.frame(
  Observation   = seq_len(imputed_n),
  Leverage      = imputed_lev,
  DFFITS        = imputed_dffits,
  CooksDistance = imputed_cooks,
  imputed_dfbetas    # one column per coefficient
)

# 3. Thresholds -------------------------------------------------------------

imputed_lev_threshold     <- 2 * imputed_p / imputed_n
imputed_dffits_threshold  <- 2 * sqrt(imputed_p / imputed_n)
imputed_cooks_threshold   <- 4 / imputed_n
imputed_dfbetas_threshold <- 2 / sqrt(imputed_n)

# 4. DFBETAS violation counts -----------------------------------------------

imputed_dfbetas_violation_count <- apply(
  abs(imputed_dfbetas) > imputed_dfbetas_threshold,
```

```
  1,
  sum
)

# 5. Total violation score ------------------------------------------------

imputed_total_violation_score <-
  (imputed_lev > imputed_lev_threshold) +
  (abs(imputed_dffits) > imputed_dffits_threshold) +
  (imputed_cooks > imputed_cooks_threshold) +
  imputed_dfbetas_violation_count

# 6. Final influence dataframe --------------------------------------------

imputed_final_df <- cbind(
  imputed_influence_df,
  DFBETAS_Violation_Count = imputed_dfbetas_violation_count,
  TotalViolations         = imputed_total_violation_score
)

# 7. Subset: keep only observations with  1 violation ----------------------

imputed_violators_df <- subset(imputed_final_df, TotalViolations > 0)

imputed_violators_non_leverage <- imputed_final_df %>%
  filter(
    abs(DFFITS) > imputed_dffits_threshold |
      CooksDistance > imputed_cooks_threshold |
      DFBETAS_Violation_Count > 0
  ) %>%
  arrange(desc(TotalViolations))

# 8. Row indices to remove for the imputed model --------------------------

imputed_remove_idx <- imputed_violators_non_leverage %>%
  filter(TotalViolations >= 10) %>%
  pull(Observation)

# 9. Identify which of these correspond to imputed rows in the original ----
# (indices of rows that were imputed in the original df_copd, restricted to non-missing pct_em
imputed_original_idx <- which(!complete.cases(df_copd[!is.na(df_copd$pct_emphysema), ]))

imputed_is_imputed <- imputed_remove_idx %in% imputed_original_idx
percent_violation_imputed <- mean(imputed_is_imputed)
# percent_violation_imputed is between 0 and 1

# 10. Refit model without influential observations ------------------------
```

```r
imputed_model_t3 <- imputed[-imputed_remove_idx, ] %>%
  slice( -c(4071,4072) )   %>%
  lm(
    pct_emphysema ~
      O2_hours_day +
      hay_fever +
      emphysema +
      copd +
      CigPerDaySmokAvg +
      Duration_Smoking +
      smoking_status +
      functional_residual_capacity +
      pct_gastrapping +
      insp_meanatt + I(insp_meanatt^2) +
      exp_meanatt + I(exp_meanatt^2) +
      FEV1_FVC_ratio +
      FVC,
    data = .
  )
```

Full vs. Reduced model

In the original study, we used multiple linear regression (MLR) to draw inferences about how various predictors relate to our outcome of interest, demonstrating that considering many variables simultaneously can provide a deeper understanding of the underlying relationships. However, this naturally raises an important question: **how much better is our MLR model compared to the best possible simple linear regression (SLR) using only one predictor?** By comparing the explanatory power of the full MLR model against the strongest single-predictor SLR, we can evaluate how much additional insight is gained by modeling the joint effects of multiple variables rather than relying on a single factor alone. This comparison helps clarify whether the complexity of MLR is justified by a meaningful improvement in predictive or inferential performance.

To evaluate how much predictive value is gained by using the full multiple linear regression (MLR) model instead of the best simple linear regression (SLR), we compared the SLR using **FEV1_FVC_ratio** as the lone predictor against the full model that included all selected variables. The SLR results show that FEV1_FVC_ratio alone explains a substantial portion of the variability in pct_emphysema, with an adjusted R^2 of **0.4646** and a residual standard error (RSE) of **0.2734**. The model is highly significant (F = **4736**, p < 2e–16), confirming that FEV1_FVC_ratio is a strong individual predictor. However, when we compare this to the full MLR model, the improvement is dramatic. The ANOVA comparison shows a reduction in residual sum of squares from **407.85** in the SLR to **106.65** in the full model. The F-test statistic of **1024.3** with a p-value < 2e–16 indicates that the full model provides a significantly better fit than the SLR. This means the additional predictors collectively explain substantially more variation in pct_emphysema than FEV1_FVC_ratio alone. Although the SLR performs reasonably well for a one-variable model, the full MLR model captures far more of the underlying structure, leading to improved accuracy and reduced error.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	2.424	0.01847	131.3	0
FEV1_FVC_ratio	-1.804	0.02621	-68.82	0

Table 10: Fitting linear model: pct_emphysema ~ FEV1_FVC_ratio

Observations	Residual Std. Error	R^2	Adjusted R^2
5457	0.2734	0.4647	0.4646

Table 11: Analysis of Variance Table

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
5455	407.9	NA	NA	NA	NA
5440	106.6	15	301.2	1024	0