

Respiratory Rate Determination by ECG Monitoring

Stephen Su, 844503, COMP90072

1 Introduction

1.1 Background

Pneumonia is the deadliest infectious disease to children under the age of 5, responsible for 740,180 deaths under 5 as of 2019 (World Health Organisation, 2022). The worsening symptoms of pneumonia can rapidly become life-threatening for young children if not treated on time. Nevertheless, the early stage of pneumonia with young children is often confused with other less serious respiratory disease with similar symptoms. On the other hand, a medical institution cannot provide comprehensive monitoring to all its patients for all potential symptoms of pneumonia due to limitation of resources. Thus it is of interest to develop light-weight, non-evasive methods for detecting the most common symptoms of pneumonia, as an early warning to serve as an indication of a need for further medical interventions.

In a research outlined in Shamo'on et al. (2004), tachypnoea (average respiratory rate > 50 per minute) is one of the critical indicators of pneumonia in young children. However, the high volatility of breath rate in children poses a major challenge for manual counting that is both time-consuming and prone to error. In contrary, automated respiratory monitoring often involves machine-measurement of chest/abdomen movement or nasal airflow that are resource-costly and possibly invasive such that the discomfort from wearing the equipment might adversely affect its accuracy.

A potential solution to the problem above arises from the phenomenon of *respiratory sinus arrhythmia*, such that the instantaneous heart rates increase with inspiration and decrease during expiration (Larsen et al., 2010), by monitoring the periodicity of variation in the heart rate, it may be possible to determine one's respiratory rate with a simple, non-evasive method, such as by a pulse oximeter that is resource-friendly. As such, this project attempts to develop an efficient *predictive* algorithm to accurately and precisely determine respiratory rate from heart rate, as well as briefly discuss its limitations and direction for future works.

1.2 The data

The *apnea-ECG database* (Penzel et al., 2000) on *PhysioNet* (Goldberger et al., 2000) provides sufficiently large data sets of approximately eight hours long records of electrocardiogram and respiratory signals for eight subjects in a study for *sleep apnea*. The eight records consist of 100Hz signal from the electrocardiogram (ECG), respiratory effort from chest/abdomen movements, nasal airflow, and blood O_2 saturation over time.

A major concern of using such data is the misalignment of objective for the researches. Study for apnea typically involves subjects with airway obstruction, a factor that must be considered for prior to building our model. Under the presumption that, subjects in our data might sometimes stop breathing, the response variable of interest should be robust from the presence of apnea. Therefore, the model will focus on exploring the algorithm for determining the respiratory rate from the ECG signal as a predictor for the respiratory rate implied by chest movements, assuming that for living humans, respiratory efforts are always present.

This study will divide records from the eight subjects into two sets:

- Subjects a01, a02, a03, a04 and b01 form the *training set*.
- Subjects c01, c02 and c03 form the *test set* for evaluation.

2 Methodologies and implementation

2.1 Data wrangling

2.1.1 Processing ECG signals

The ECG signal does not provide the heart rate directly. Instead, it is a time series of varying electric potential that controls the rhythm of the contraction and relaxation of the heart muscle, with a magnitude of approximately 0.5mV in absolute value. Such a signal is often masked by unwanted noise, such as electric signals from pectoral muscle movement, creating a challenge in computing heart rate from the ECG signal.



Figure 1: Left: sudden spike in noise; right: frequency spectrum of the ECG signal

Algorithm 1 5-20Hz band-pass filter with fast Fourier transform

- 1: Take $\mathbf{x} \in \mathbb{R}^n$ as the 100-Hz input
 - 2: Obtain \mathbf{y} by fast Fourier transform on \mathbf{x}
 - 3: **for** $t = 1, \dots, n$ **do**
 - 4: $y_t \leftarrow \sum_{k=1}^n x_k \exp(-2\pi i(k-1)(t-1)/n)$
 - 5: **end for**
 - 6: Zero corresponding (Nyquist) frequency bins for 0-5, 20-50Hz
 - 7: **for** $t = 1, \dots, n \wedge (t \in (0.2n, 0.8n) \vee t > 0.95n \vee t < 0.05n)$ **do**
 - 8: $y_t \leftarrow 0 + 0i$
 - 9: **end for**
 - 10: Obtain \mathbf{z} by normalised inverse fast Fourier transform on \mathbf{y}
 - 11: **for** $t = 1, \dots, n$ **do**
 - 12: $z_t \leftarrow \sum_{k=1}^n y_k \exp(2\pi i(k-1)(t-1)/n)/n$
 - 13: **end for**
 - 14: **return** $\text{Re}(\mathbf{z}) \in \mathbb{R}^n$ as the 100-Hz output
-

2.1.2 Heart rate derivation from ECG signals

Algorithm 2 R peak timestamping with phase correction

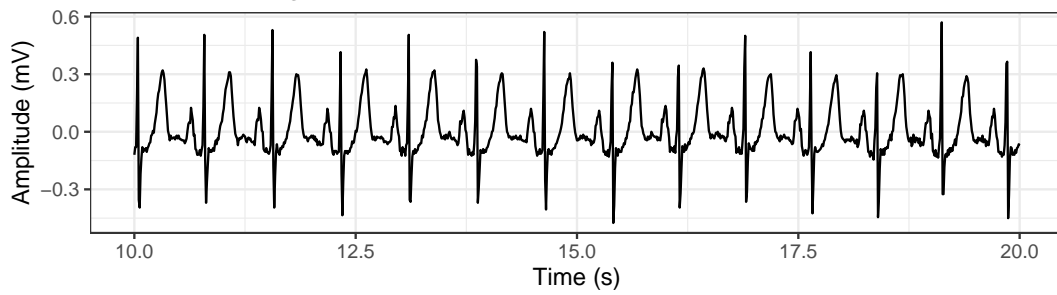
```
1: Take the original  $\mathbf{x} \in \mathbb{R}^n$  and the filtered  $\mathbf{z} \in \mathbb{R}^n$  as the 100-Hz inputs
2: Set moving threshold  $\mathbf{b}$  as half of rolling maximum
3:  $(b_t | t = 1, \dots, 100) \leftarrow (+\infty)_{\times 100}$ 
4: for  $t = 101, \dots, n$  do
5:    $b_t \leftarrow \frac{1}{2} \max\{z_{t-100}, \dots, z_t\}$ 
6: end for
7: Create thresholding indicator  $\iota^{(1)}$ 
8: for  $t = 1, \dots, n$  do
9:    $\iota_t^{(1)} \leftarrow I(z_t > \max\{b_t, F_Z^{-1}(0.95)\})$ 
10: end for
11: Create concave-peak indicator  $\iota^{(2)}$ 
12: for  $t = 1, n$  do
13:    $\iota_t^{(2)} \leftarrow 0$ 
14: end for
15: for  $t = 2, \dots, n-1$  do
16:    $\iota_t^{(2)} \leftarrow I[I(z_{t+1} - z_t > 0) - I(z_t - z_{t-1} > 0) = -1]$ 
17: end for
18: Preliminary timestamps for R peak  $\mathbf{p} \leftarrow (t = 1, \dots, n | \iota_t^{(1)} \wedge \iota_t^{(2)} \text{ is True})$ 
19: Remove abnormal timestamps in  $\mathbf{p}$  with R-R intervals of  $< 300\text{ms}$ 
20: Correct phase-shifts caused by fast Fourier transform
21: for  $t \in \{\mathbf{p}\}$  do
22:    $p | p = t \leftarrow \operatorname{argmax}_{i=t-4, \dots, t, \dots, t+4} x_i$ 
23: end for
24: return  $\mathbf{p}$  the timestamps of the R peak
```

Algorithm 3 Deriving instantaneous heart rates from R peak timestamps

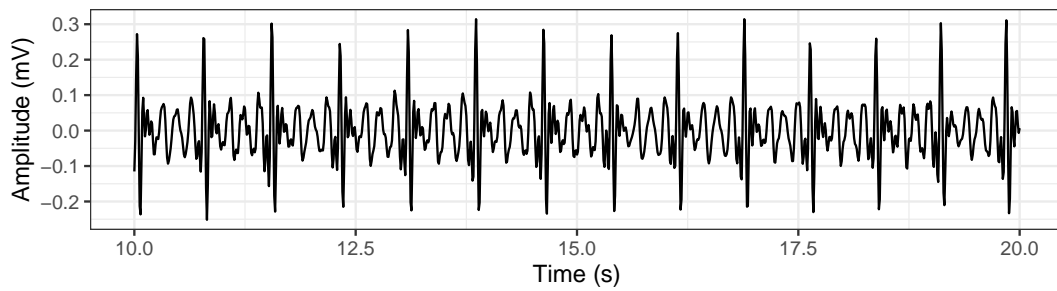
```
1: Take  $\mathbf{p}$  the timestamps of the R peak as the input, and  $\nu$  as an argument for the frequency unit of  $\mathbf{p}$ 
2:  $m \leftarrow \dim(\mathbf{p})$  dimension of  $\mathbf{p}$ 
3: Compute the R-R intervals  $\mathbf{d}$ 
4:  $(d_t | t = 1, \dots, p_1) \leftarrow (\text{NA})_{\times p_1}$ 
5:  $(d_t | t = p_1 + 1, \dots, p_m) \leftarrow ((p_2 - p_1)_{\times (p_2 - p_1)}, (p_3 - p_2)_{\times (p_3 - p_2)}, \dots, (p_m - p_{m-1})_{\times (p_m - p_{m-1})})^\top$ 
6: Convert R-R intervals into heart rates per minute  $\mathbf{h}$ 
7:  $\mathbf{h} \leftarrow 60 \cdot \nu / \mathbf{d}$ 
8: return  $\mathbf{h}$  the heart rates per minute at frequency  $\nu$ 
```

2.1.3 Visualising ECG signal processing

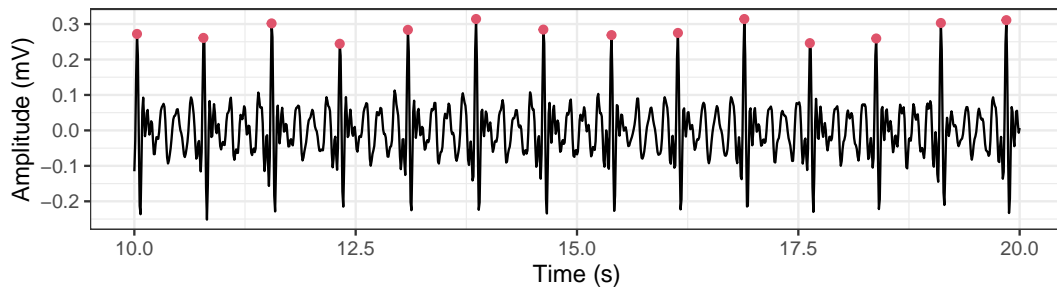
(1) Raw ECG signal



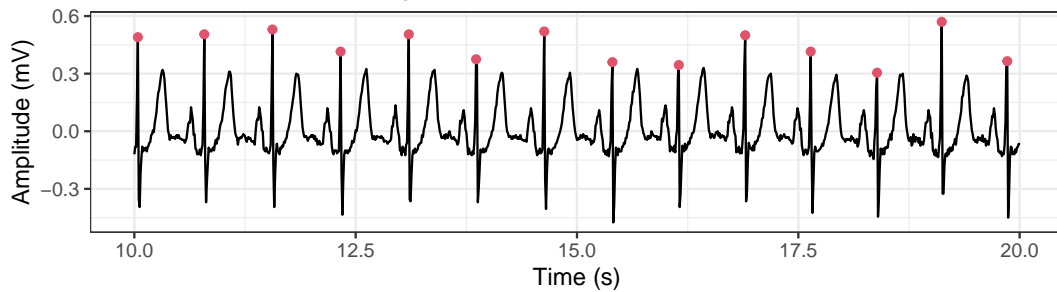
(2) FFT filter to suppress P and T waves



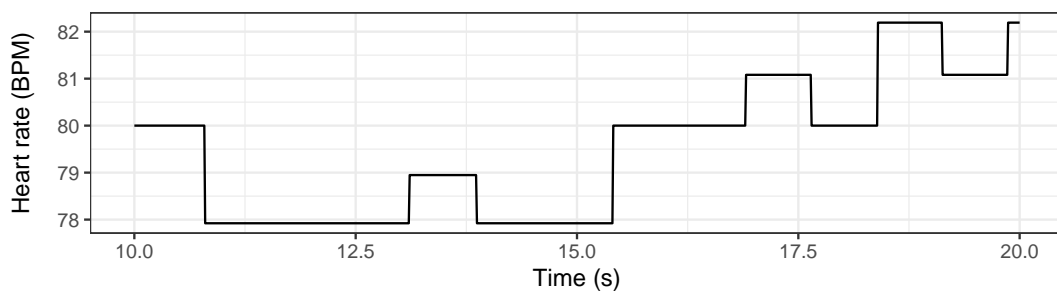
(3) Find R peak on filtered signal



(4) Map R peak to raw signal with phase correction



(5) Derive heart rate from R-R intervals



2.2 Modelling rate of breathing by heart rate

2.2.1 The smoothing spline crossing algorithm

Algorithm 4 Smoothing spline crossing for counting cycles of nonstationary pseudosinusoidal oscillations, for varying heart rates/chest movements

- 1: Take a time series vector \mathbf{a} at any frequency, and the optional argument λ the regularisation parameter
 - 2: $\mathbf{t} \leftarrow (1, \dots, \dim(\mathbf{a}))^\top$ as the indexing vector
 - 3: Construct B-spline bases \mathbf{B} derived from natural cubic spline bases about all possible knots of \mathbf{t}
 - 4: Penalty matrix $\mathbf{\Omega} \leftarrow \{\omega_{jk} = \int_T B_j''(s)B_k''(s)ds | B_j(t_i) = (\mathbf{B})_{ij}\}$
 - 5: **if** argument λ is missing **then**
 - 6: $\lambda \leftarrow \operatorname{argmin}_{\lambda > 0} (1 - \operatorname{tr}(\mathbf{B}(\mathbf{B}^\top \mathbf{B} + \lambda \mathbf{\Omega})^{-1} \mathbf{B}^\top)) / \dim(\mathbf{a})^{-2} \|(\mathbf{I} - \mathbf{B}(\mathbf{B}^\top \mathbf{B} + \lambda \mathbf{\Omega})^{-1} \mathbf{B}^\top) \mathbf{a}\|^2$
 - 7: **end if**
 - 8: $\hat{\mathbf{a}} \leftarrow \mathbf{B}(\mathbf{B}^\top \mathbf{B} + \lambda \mathbf{\Omega})^{-1} \mathbf{B}^\top \mathbf{a}$
 - 9: Count the number of times the curves (\mathbf{t}, \mathbf{a}) and $(\mathbf{t}, \hat{\mathbf{a}})$ cross (equivalent to zero-crossing for $(\mathbf{t}, \mathbf{a} - \hat{\mathbf{a}})$)
 - 10: **return** half of the count obtained above as the approximated cycles
-

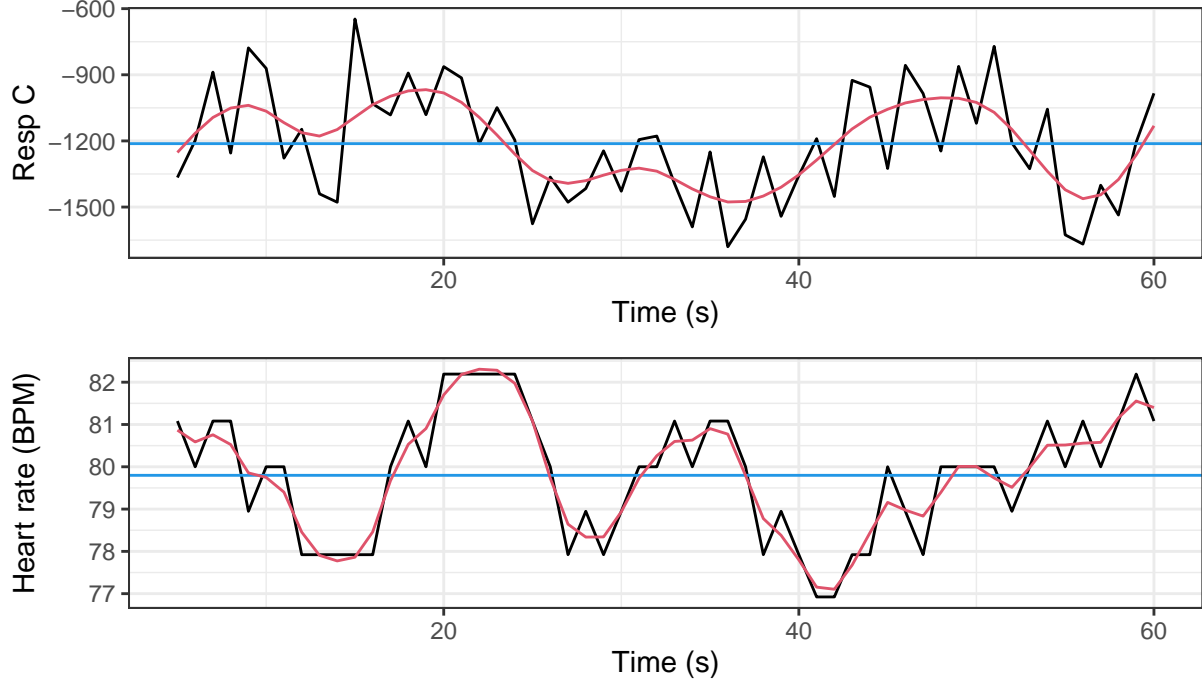


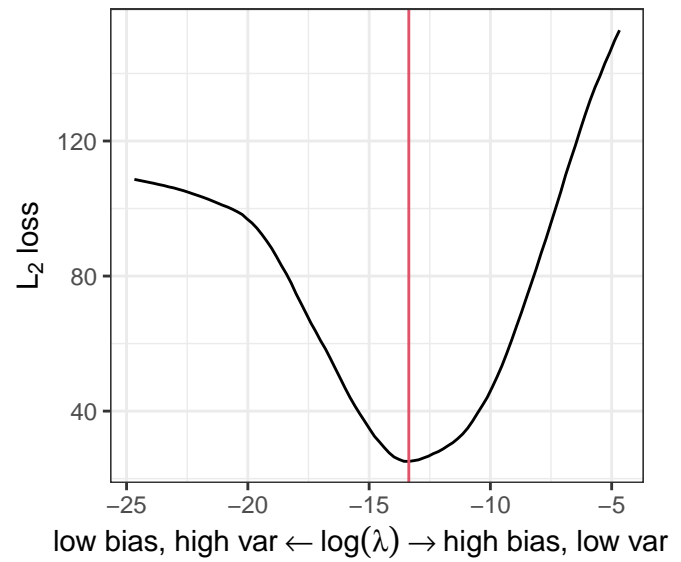
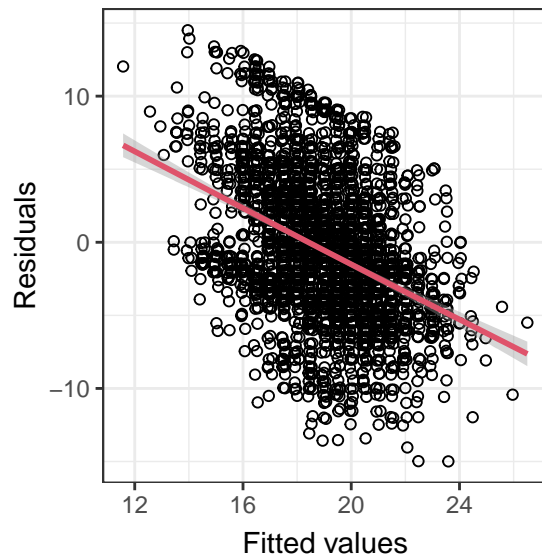
Figure 2: Time series of chest movement (top) and heart rates (bottom) at 1Hz for 1 minute. Blue: naive zero-crossing gives a count of 11.5 and 6 breaths; red: smoothing spline crossing gives a count of 19.5 and 16.5 breaths. We can clearly see that smoothing spline performs much better in counting breaths.

2.2.2 Model training

Algorithm 5 Deriving breath rate from heart rate based on chest movements

- 1: Take \mathbf{h} the heart rates per minute and \mathbf{c} the chest movements signal (either at 100Hz or down-sampled) where both time series have aligned indices and identical frequency; and optional argument λ_{opt} (only in model evaluation with test data)
 - 2: For both \mathbf{h} and \mathbf{c} , remove elements of index where $c_t = \text{NA}$
 - 3: **for** each record of all the subjects **do**
 - 4: Partition each of \mathbf{h} and \mathbf{c} into segments of 60-second data
 - 5: **end for**
 - 6: Drop the remainder (last segment), which is less than 60 seconds
 - 7: **if** \mathbf{h} and \mathbf{c} for more than one subject is input **then**
 - 8: Bind \mathbf{h} and \mathbf{c} of all subjects together
 - 9: **end if**
 - 10: **for** each 60-second segment of \mathbf{c} **do**
 - 11: Pass each minute of \mathbf{c} into Algorithm 4 (the smoothing spline crossing, without specifying λ) to compute the true rate of breath based on chest movements, and output $\rightarrow \mathbf{r}$
 - 12: **end for**
 - 13: **if** argument λ_{opt} is missing **then**
 - 14: $\lambda_{\text{opt}} \leftarrow \text{argmin}_{\lambda > 0} \|\mathbf{r} - \hat{f}(\mathbf{h}; \lambda)\|^2$ where \hat{f} is the Algorithm 4: applied on each minute of \mathbf{h} given λ
 - 15: **end if**
 - 16: $\hat{\lambda}_{\text{opt}} \leftarrow \lambda_{\text{opt}}$: if its value is obtained by Step 14 $\hat{\lambda}_{\text{opt}}$ is the estimate of λ_{opt} derived from the training data; otherwise, if λ_{opt} is passed as an argument, then the purpose of this algorithm is model testing
 - 17: **for** each 60-second segment of \mathbf{h} **do**
 - 18: Pass each minute of \mathbf{h} into Algorithm 4 (with argument of $\lambda = \hat{\lambda}_{\text{opt}}$) to compute the derived rate of breath from heart rate, and output $\rightarrow \hat{\mathbf{r}}$ which is the model's estimate of \mathbf{r} : the chest rate of breath
 - 19: **end for**
 - 20: **return** data matrix $(\mathbf{r}, \hat{\mathbf{r}})$, a bi-variate time series (minutely), $\mathbf{r} :=$ "count of breaths by chest movements (observations)" and $\hat{\mathbf{r}} :=$ "number of breaths derived by cardio-activities of that minute (fitted values)", and most importantly $\hat{\lambda}_{\text{opt}}$ the trained parameter for our final model!
-

2.3 Model diagnostics



2.4 Model evaluation

3 Appendix: workflow, code and output

3.1 Source code

The source codes for the functions used in the workflow are in the .R files at [/R](#).

3.2 Data wrangling and signal processing

```
training_set <- c("a01", "a02", "a03", "a04", "b01")
test_set <- c("c01", "c02", "c03")
hr <- fuse_data(map(
  sprintf("../data-bin/%s.dat", training_set),
  function(ecg_file) {
    ecg_file |>
      read_ecg() |>
      find_r_peaks() |>
      frequency() |>
      down_sample()
  }
))
resp <- fuse_data(map(
  sprintf("../data-bin/%sr.dat", training_set),
  function(resp_file) down_sample(read_resp(resp_file))
))
hr_test <- fuse_data(map(
  sprintf("../data-bin/%s.dat", test_set),
  function(ecg_file) {
    ecg_file |>
      read_ecg() |>
      find_r_peaks() |>
      frequency() |>
      down_sample()
  }
))
resp_test <- fuse_data(map(
  sprintf("../data-bin/%sr.dat", test_set),
  function(resp_file) down_sample(read_resp(resp_file))
))
write_rds(hr, "../R/hr.rds")
write_rds(resp, "../R/resp.rds")
write_rds(hr_test, "../R/hr-test.rds")
write_rds(resp_test, "../R/resp-test.rds")
```

3.3 Model training

```
hr <- read_rds("../R/hr.rds")
resp <- read_rds("../R/resp.rds")
resp_df <- resp_dataset(hr, resp)
```

```
attributes(resp_df)$opt_lambda
```

```
#>   opt_lambda  
#> 1.571851e-06
```

```
resp_df
```

```
#> # A tibble: 2,526 x 2  
#>   breath_chest breath_ecg  
#> *      <dbl>      <dbl>  
#> 1         19.5         15  
#> 2         22.5         16  
#> 3         23.5        17.5  
#> 4         20.5         18  
#> 5         20.5         18  
#> 6          23         15  
#> 7          16         18  
#> 8          20        21.5  
#> 9          20        17.5  
#> 10         18        16.5  
#> # i 2,516 more rows
```

```
summary(resp_df)
```

```
#>   breath_chest   breath_ecg  
#> Min.   : 5.50   Min.   :11.50  
#> 1st Qu.:15.00   1st Qu.:17.50  
#> Median :18.50   Median :19.00  
#> Mean   :18.46   Mean   :18.96  
#> 3rd Qu.:21.38   3rd Qu.:20.50  
#> Max.   :29.00   Max.   :26.50
```

3.4 Model diagnostics

```
c("R-F cor" = with(resp_df, cor(breath_chest - breath_ecg, breath_ecg)))
```

```
#>   R-F cor  
#> -0.3976376
```

```
test <- lm(breath_chest ~ 0 + breath_ecg, resp_df)  
c("p-value" = unname(pchisq(  
  sum(resp_df$breath_ecg^2) * (coef(test) - 1)^2 *  
    (nrow(resp_df) - 1) / sum(residuals(test)^2), 1,  
  lower.tail = FALSE  
)))
```

```
#>   p-value  
#> 2.983174e-13
```

```
confint(test)
```

```
#>                2.5 %    97.5 %  
#> breath_ecg 0.9521173 0.9724053
```

3.5 Model evaluation

```
hr_test <- read_rds("../R/hr-test.rds")  
resp_test <- read_rds("../R/resp-test.rds")  
resp_dt <- resp_dataset(hr_test, resp_test, 1.571851e-06)  
c("RMSEP" = with(resp_dt, sqrt(mean((breath_chest - breath_ecg)^2))))
```

```
#>    RMSEP  
#> 5.489861
```

References

- Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., & Stanley, H. E. (2000). Physiobank, physiotoolkit, and physionet. *Circulation*, *101*(23), e215–e220.
- Larsen, P. D., Tzeng, Y. C., Sin, P. Y. W., & Galletly, D. C. (2010). Respiratory sinus arrhythmia in conscious humans during spontaneous respiration. *Respiratory Physiology & Neurobiology*, *174*(1–2), 111–118. <https://doi.org/10.1016/j.resp.2010.04.021>
- Penzel, T., Moody, G. B., Mark, R. G., Goldberger, A. L., & Peter, J. H. (2000). The apnea-ecg database. *Computers in Cardiology 2000*, 255–258.
- Shamo'on, H., Hawamdah, A., Haddadin, R., & Jmeian, S. (2004). Detection of pneumonia among children under six years by clinical evaluation. *East Mediterr Health J*, *10*(4-5), 482–487.
- World Health Organisation. (2022). *Pneumonia in children*. <https://www.who.int/news-room/fact-sheets/detail/pneumonia>