Respiratory Rate Determination by ECG Signals

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.

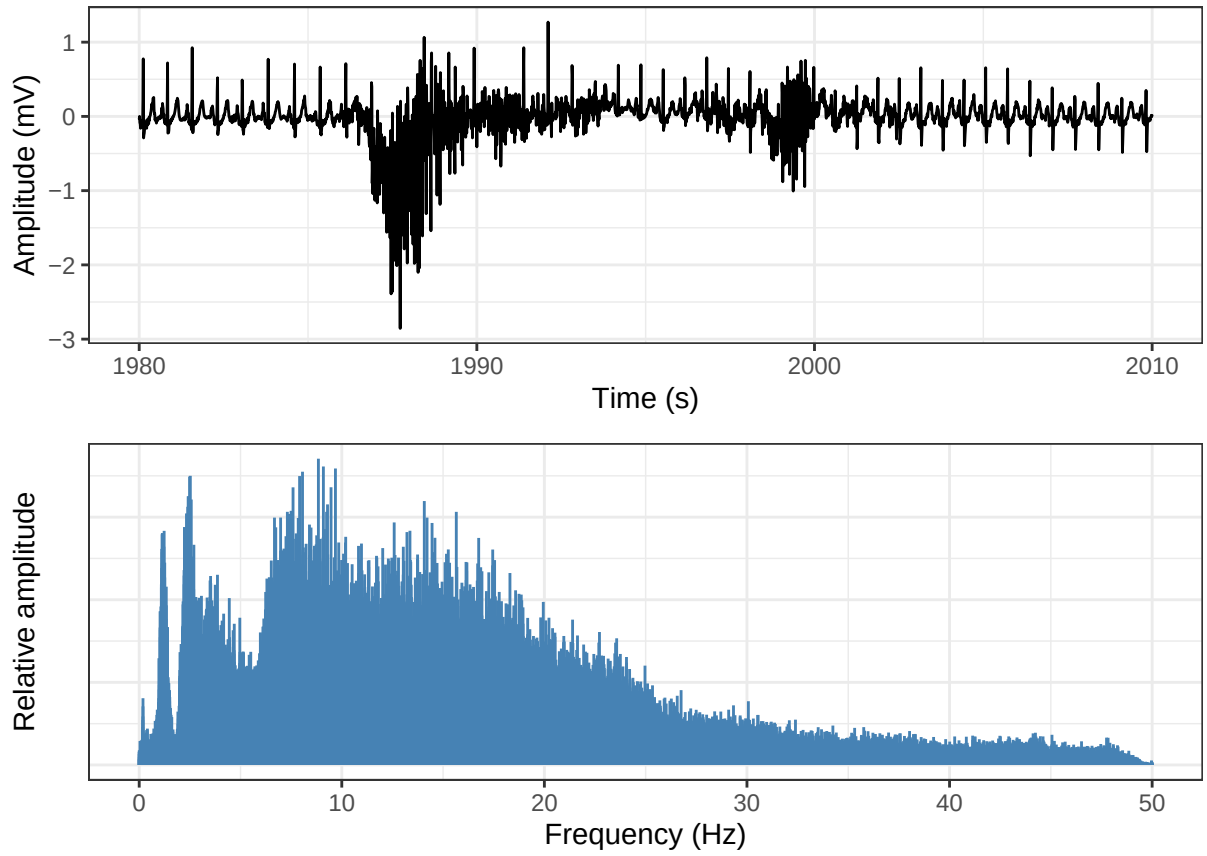


Figure 1: Top: sudden spike in noise; bottom: frequency spectrum of the ECG signal

Cardio-activities are reflected in the ECG signal as time series of QRS complex as counts for complete heart beats (Goldberger et al., 2017), where the instantaneous heart rates are determined by the interval of consecutive R-peaks, characterised by a positive crest in electric potential with a frequency of approximately 10–20Hz, and are distinguishable from P and T peaks with frequencies mostly below 5Hz.

Apart from muscle-induced electric noise, another challenge arises from abnormally peaked P and T waves from a variety of heart conditions (for example, abnormally peaked T waves observed on subject a02). Upon de-noising the ECG signal, it is also needed to suppress the peaks for P and T waves. Otherwise, they become less distinguishable from the R peaks amplitude-wise and might be mistakenly flagged as R peaks.

Intuitively, the simplest method for reducing the noise and suppressing P and T peaks is to filter out their typical frequencies. With observed frequencies of noise from the frequency spectrum mostly above 20Hz,

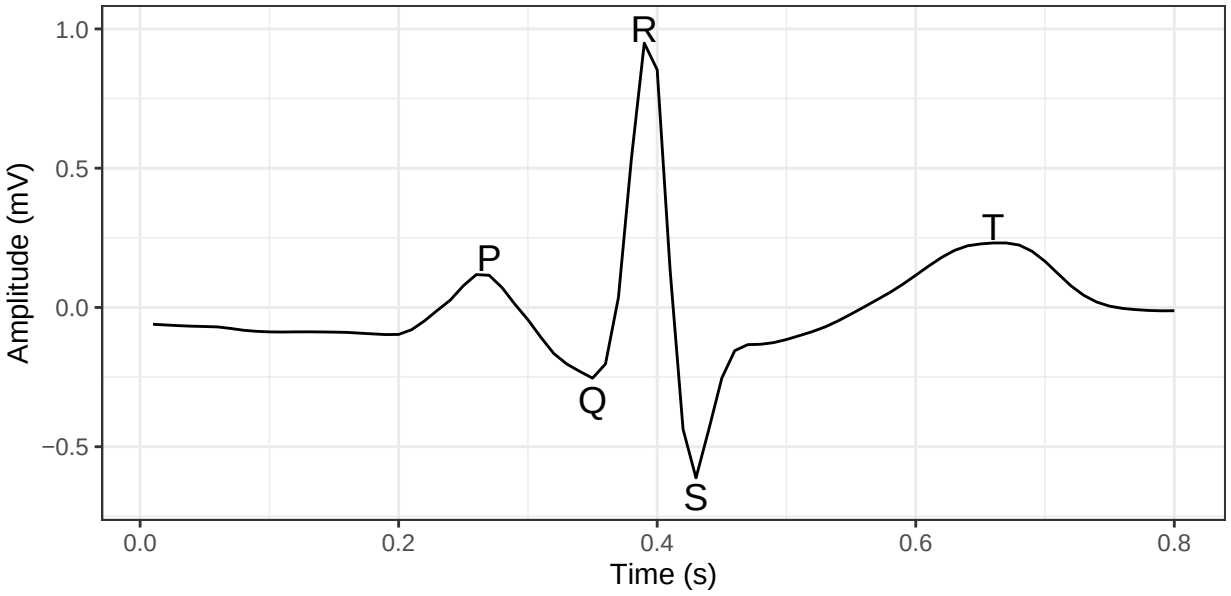


Figure 2: An example QRS complex from the ECG signal for one complete heart beat

and the characteristic frequencies of P and T waves being below 5Hz, we may utilise the Discrete Fourier Transform (DFT) (Cooley & Tukey, 1965), to create a band-pass filter, retaining only 5–20Hz signals from the ECG data. The DFT transformed signal compromises the Nyquist frequency bins (with its range being half of sampling frequency) (Nyquist, 1928) corresponding to each of the harmonic. The band-pass filtered signal is then obtained by applying the (normalised) inverse Discrete Fourier Transform to the DFT transformed signal, with zeroed frequency bins for 0-5 and 20-50Hz. For computational efficiency, the Discrete Fourier Transform is implemented by R (R Core Team, 2023) with the Fast Fourier Transform (FFT) algorithm (Singleton, 1979) using the already available `fft()` function (Becker et al., 1988).

It is worth noting that, the DFT approach to band-pass filtering as per outlined above has not accounted for the modulation effect from Fourier Transform, and will likely cause phase shifts in the de-noised signals. Nevertheless, upon thorough inspection, the observed phase error in the filtered output are mostly within 10–20ms, and easily correctable with local peak-matching, which is discussed in the following paragraphs.

Note: most "for loops" in the pseudocode algorithms in this report are implicitly implemented by vectorisation.

Algorithm 1 5–20Hz band-pass filter with Discrete Fourier Transform

Take $\mathbf{x} \in \mathbb{R}^n$ as the 100-Hz input
Obtain $\mathbf{y}$ by Discrete Fourier Transform on $\mathbf{x}$:
for $t = 1, ..., n$ **do**
    $y_t \leftarrow \sum_{k=1}^{n} x_k \exp(-2\pi i(k-1)(t-1)/n)$
end for
Zero corresponding (Nyquist) frequency bins for 0-5, 20-50Hz:
for $t = 1, ..., n \wedge (t \in (0.2n, 0.8n) \vee t > 0.95n \vee t < 0.05n)$ **do**
    $y_t \leftarrow 0 + 0i$
end for
Obtain $\mathbf{z}$ by normalised inverse Discrete Fourier Transform on $\mathbf{y}$:
for $t = 1, ..., n$ **do**
    $z_t \leftarrow \sum_{k=1}^{n} y_k \exp(2\pi i(k-1)(t-1)/n)/n$
end for
return $\mathrm{Re}(\mathbf{z}) \in \mathbb{R}^n$ as the 100-Hz output

With the de-noised ECG signal, a special algorithm for timestamping the R peaks is developed based on logical check for local maxima and double thresholding. For the amplitudes of P and T peaks are suppressed in the filtered signal, the local maxima above certain threshold in the series are the R peaks. For a series of discrete-time signal, the local maxima are indicated by whenever the signs of the first difference change from *positive* to *negative*. Also, as the R peaks are more than triple the amplitude of the DFT suppressed P and T peaks, all local maxima with an amplitude of at least half (in prudence) the amplitude of previous peak are indicated as the R peaks. Upon implementation, the threshold is set at half of the rolling maxima for the last 101 ticks (1.01s). For extra caution to unlikely events when there is no heart beat in the past 1.01 seconds, a second threshold, the 95% empirical quantile of amplitude of the signal, is imposed. All time-points in the filtered ECG signal that are deemed as local maxima also with amplitude above the two thresholds are stamped as R peaks. The algorithm is fast even for long ECG signal.

There are, however, issues that cannot be dealt with via aforementioned algorithms. While the algorithms are robust against moderate noise with filtering, they will unlikely function in the presence of extreme noise with amplitude exceeding the original signal. Under such extreme noise, which is rarely seen, the stamped R peaks would manifest small clusters of abnormally short R-R intervals, typically below 300ms. Therefore, an additional check is implemented to identify and remove all the abnormal timestamps. Another issue with the algorithm is the phase shift induced by Fourier Transform. Without taking care for the modulation harmonics, the timestamps obtained from the filtered signal are ±10–20ms away from the true observed R peaks in the original signal. Fortunately, the phase errors are small and can be easily corrected by searching for the existence of another peak around the neighbours in the original signal.

Algorithm 2 R peak timestamping with phase correction

Take the original $\mathbf{x} \in \mathbb{R}^n$ and the filtered $\mathbf{z} \in \mathbb{R}^n$ as the 100-Hz inputs
Set moving threshold $\mathbf{b}$ as half of rolling maxima:
$(b_t | t = 1, ..., 100) \leftarrow (+\infty)_{\times 100}$ (rolling maxima is undefined for the first second, pad using $+\infty$)
for $t = 101, ..., n$ **do**
    $b_t \leftarrow \frac{1}{2} \max\{z_{t-100}, ..., z_t\}$
end for
Create thresholding Boolean $\boldsymbol{\iota}^{(1)}$:
for $t = 1, ..., n$ **do**
    $\iota_t^{(1)} \leftarrow I(z_t > \max\{b_t, F_Z^{-1}(0.95)\})$
end for
Create local maxima Boolean $\boldsymbol{\iota}^{(2)}$:
for $t = 1, n$ **do**
    $\iota_t^{(2)} \leftarrow 0$ (local maxima are undefined for the boundaries)
end for
for $t = 2, ..., n-1$ **do**
    $\iota_t^{(2)} \leftarrow I[I(z_{t+1} - z_t > 0) - I(z_t - z_{t-1} > 0) = -1]$
end for
Preliminary timestamps for R peak $\mathbf{p} \leftarrow (t = 1, ..., n | \iota_t^{(1)} \wedge \iota_t^{(2)} = 1$ is True)
Remove abnormal timestamps in $\mathbf{p}$ with R-R intervals of < 300ms
Correct phase-shifts caused by discrete Fourier Transform:
for $t \in \{\mathbf{p}\}$ **do**
    $p | p = t \leftarrow \arg\max_{i=t-4,...,t,...,t+4} x_i$
end for
return $\mathbf{p}$ the timestamps of the R peak

2.1.2 Heart rate derivation from ECG signals

Once all the time points of R peaks are identified, we have cleared all obstructions to determining the heart rate. At any point in time within the series of ECG signal, the instantaneous heart rate is determined by interval between the subsequent and consequent R peaks. While one might explicitly loop over all the 5×3 million data points in ECG to obtain the series of intervals between R peaks, the process can be more efficient via vectorisation. It can be shown, with an example, that the derived series of R-R interval as a function of series of timestamps of the R peaks, is equivalent to repeating the differenced timestamps each for numbers of times equalled to the differenced timestamps themselves.

For example, the R peaks are on the 3rd, 7th, 10th and 12th time ticks (for illustration purpose only), then the series of R-R interval can be computed by the following R (R Core Team, 2023) code:

```
r_peak <- c(3, 7, 10, 12)
r_peak
```

```
#> [1]  3  7 10 12
```

```
rr_interval <- c(rep(NA, r_peak[1]), rep(diff(r_peak), diff(r_peak)))
rr_interval
```

```
#>  [1] NA NA NA  4  4  4  4  3  3  3  2  2
```

```
length(rr_interval)
```

```
#> [1] 12
```

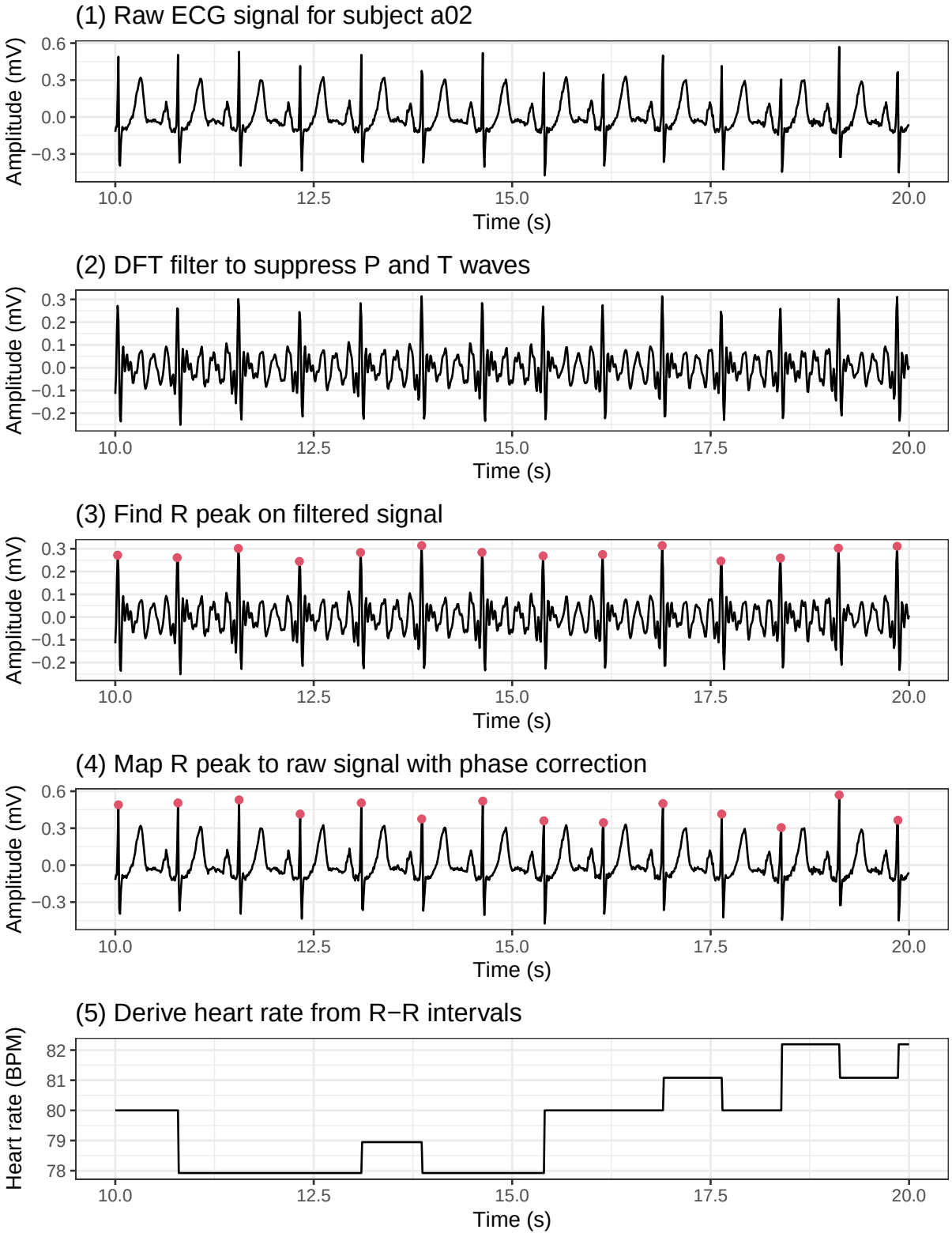
The series of instantaneous heart rate per minute is then obtainable by 60 $\times$ frequency / R-R intervals.

Algorithm 3 Deriving instantaneous heart rates from R peak timestamps

1: Take $\mathbf{p}$ the timestamps of the R peak as the input, and ν 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: $(d_t | t = 1, ..., p_1) \leftarrow (\text{NA})_{\times p_1}$ (heart rate is undefined before the first R peak)
5: $(d_t | t = p_1 + 1, ..., p_m) \leftarrow ((p_2 - p_1)_{\times (p_2 - p_1)}, (p_3 - p_2)_{\times (p_3 - p_2)}, ..., (p_m - p_{m-1})_{\times (p_m - p_{m-1})})^\top$
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 ν

The output series of instantaneous heart rate is a signal at 100Hz with stepped (quantised) values, as the resolution of the derived heart rate is limited by the sampling frequency of the ECG signal (see next page).

2.1.3 Visualising ECG signal processing



2.2 Modelling rate of breathing by heart rate

As per phenomenon of *respiratory sinus arrhythmia* described in Larsen et al. (2010), we expect to observe a periodic variation in heart rates with chest movements while breathing. Therefore, the final objective is to develop an algorithm to accurately "count" the number of respiration from both heart rate variation and chest movements, and hence construct a model to predict the rate of respiration using heart rate alone.

2.2.1 The smoothing spline crossing algorithm

An algorithm, which I name the smoothing spline crossing algorithm, was inspired by the idea of zero crossing (Giannakopoulos & Pikrakis, 2014), often used in audio processing for estimating the number of wave oscillations from the number of sign changes (when the signal has crossed the zero axis):

$$\mathrm{Z}(\mathbf{a}) \approx \frac{1}{2} \sum_{t=2}^{\dim(\mathbf{a})} |\mathrm{sgn}(a_t) - \mathrm{sgn}(a_{t-1})|$$

For series with a nonzero mean, a generalisation to zero crossing is to count the oscillations by mean crossing:

$$\mathrm{Z}(\mathbf{a}) \approx \frac{1}{2} \sum_{t=2}^{\dim(\mathbf{a})} |\mathrm{sgn}(a_t - \bar{a}) - \mathrm{sgn}(a_{t-1} - \bar{a})|$$

Nonetheless, the mean crossing estimate only works well if, and only if each oscillation almost always crosses the sample mean, e.g., an almost perfect sinusoidal wave. Yet the series of instantaneous heart rate and chest movement signals manifested oscillations with irregular cycles of nonstationary mean, such that a naive application of mean crossing will inevitably give a highly biased oscillation count. We thus need to find an alternative to sample mean for crossing for a better count of number of cycles for a nonstationary oscillation.

The smoothing spline (Green & Silverman, 1994; Hastie & Tibshirani, 1990) is a bivariate smoothing method based on regularised regression to a natural cubic spline basis, penalised by the integrated squared second derivative (which represents the roughness) of the smoothed curve, depictable as a function:

$$\hat{\mathbf{a}} = \hat{f}(\mathbf{t}; \lambda) = \underset{f(\mathbf{t})}{\arg\min} \Big\{ \sum_{t=1}^{\dim(\mathbf{a})} (a_t - f(t))^2 + \lambda \int_t f''(s)^2 ds \Big\} \text{ given any } \lambda > 0 \tag{1}$$

Where λ is the regularisation parameter controlling the penalty to the roughness of the fitted values, which effectively determines how smooth the fitted curve becomes. With *properly* selected λ, we are able to draw a better boundary for crossing, to give a better count for the oscillations, than the sample mean of the heart rates and chest movements. The proposed smoothing spline crossing method is given by:

$$\begin{aligned} \mathrm{S}(\mathbf{a}; \lambda) &\approx \frac{1}{2} \sum_{t=2}^{\dim(\mathbf{a})} |\mathrm{sgn}(a_t - \hat{a}_t) - \mathrm{sgn}(a_{t-1} - \hat{a}_{t-1})| \\ &= \frac{1}{2} \sum_{t=2}^{\dim(\mathbf{a})} |\mathrm{sgn}(a_t - \hat{f}(t; \lambda)) - \mathrm{sgn}(a_{t-1} - \hat{f}(t-1; \lambda))| \end{aligned}$$

The smoothing spline fitting for heart rates and chest movements signal is implemented using R (R Core Team, 2023) function `smooth.spline()` (Chambers & Hastie, 1992). It uses the *B-spline*-equivalent to the

natural cubic spline, a linear transformation of the spline basis to an orthonormal basis such that it gives the same smoothing curve, but numerically stabler in computation. The process of minimising the loss function in (1), numerically solving its first-order condition, is included in Algorithm 4 in matrix form.

It is noteworthy that the computational burden of the `smooth.spline()` function (Chambers & Hastie, 1992) arises from matrix computation for numerically solving the inverse of cross product for the *B-spline* basis, and such operation costs $O(\dim(\mathbf{a})^3)$ computation time. However, with our minutely-segmented approach (see the next section and Algorithm 5), number of data points in each of the segment is constant; and since the number of segments is proportional to $\dim(\mathbf{a})$, it reduces the computation time to $O(\dim(\mathbf{a}))$. Yet with over 15 million data points in the training set, the fitting process for the smoothing spline would still be unnecessarily slow, for considering the rate of breathing of an average child, a sampling frequency of 100Hz is redundantly high, for the purpose of capturing respiratory information. For this report, heart rates and chest movements signal is down-sampled to 1Hz immediately after filtering and prior to fitting. Note that, the selection of 1Hz is based on trial and not theoretically justified, but can be easily changed to another frequency if really necessary. Another practical reason for using 1Hz is that our subjects are adults. Further researches may study the adaptive selection of sampling frequencies, to best balance accuracy and efficiency.

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

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

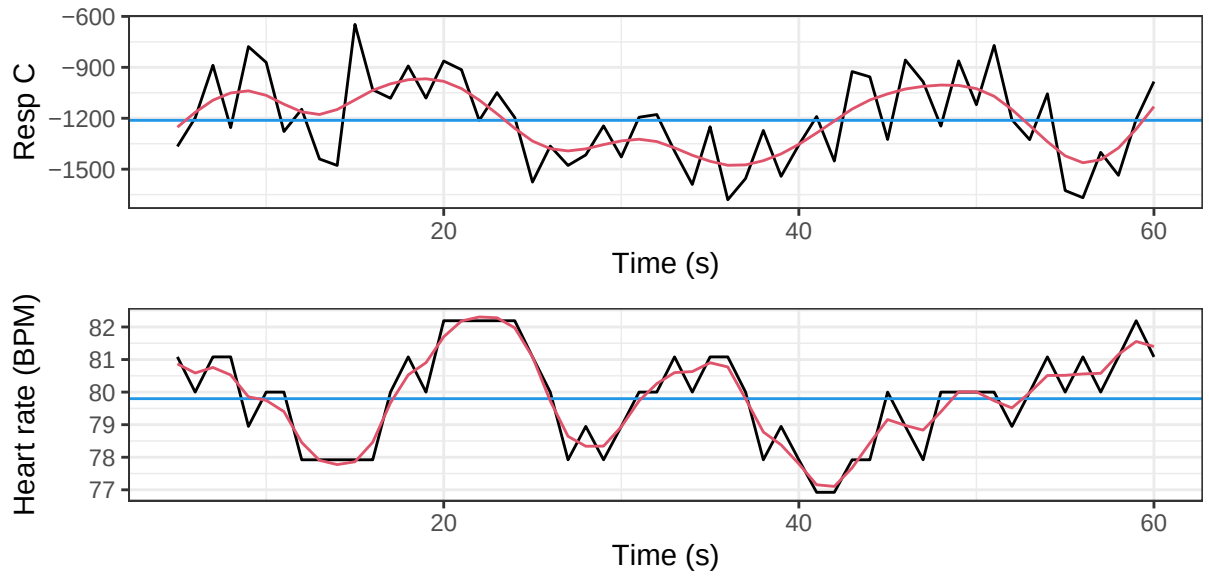


Figure 3: Time series of chest movement (top) and heart rates (bottom) at 1Hz for 1 minute. Blue: naive mean-crossing gives a count of 11.5 and 6 breaths; red: smoothing spline crossing gives a count of 19.5 and 16.5 breaths. We can clearly see that smoothing spline performs much better in counting breaths.

2.2.2 Model training

(Incomplete)

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

1: Take **h** the heart rates per minute and **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 **h** and **c**, remove elements of index where $c_t = \text{NA}$
3: **for** each record of all the subjects **do**
4: Partition each of **h** and **c** into segments of 60-second data
5: **end for**
6: Drop the remainder (last segment), which is less than 60 seconds
7: **if** **h** and **c** for more than one subject is input **then**
8: Bind **h** and **c** of all subjects together
9: **end if**
10: **for** each 60-second segment of **c** **do**
11: Pass each minute of **c** into Algorithm 4 (the smoothing spline crossing, without specifying λ) to compute the true rate of breath based on chest movements, and output $\rightarrow$ **r**
12: **end for**
13: **if** argument λ_{opt} is missing **then**
14: $\lambda_{\text{opt}} \leftarrow \arg\min_{\lambda>0} ||\mathbf{r} - \hat{f}(\mathbf{h}; \lambda)||^2$ where $\hat{f}$ is the Algorithm 4: applied on each minute of **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 **h** **do**
18: Pass each minute of **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 **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

The trained model implied that one's heart rate is useful for inferring the rate of breath, and the results are statistically significant based on the regularisation result. If there is null relationship between the heart rate and rate of breath, the trained model should have returned a *least-complex fit* such that L_2 loss function (cross-validated MSE) is minimised at large λ and is monotonically decreasing. Hence, the obvious inverted bell curve exhibited by the regularised L_2 loss with one minimum gives evidence against a *null-model fit*.

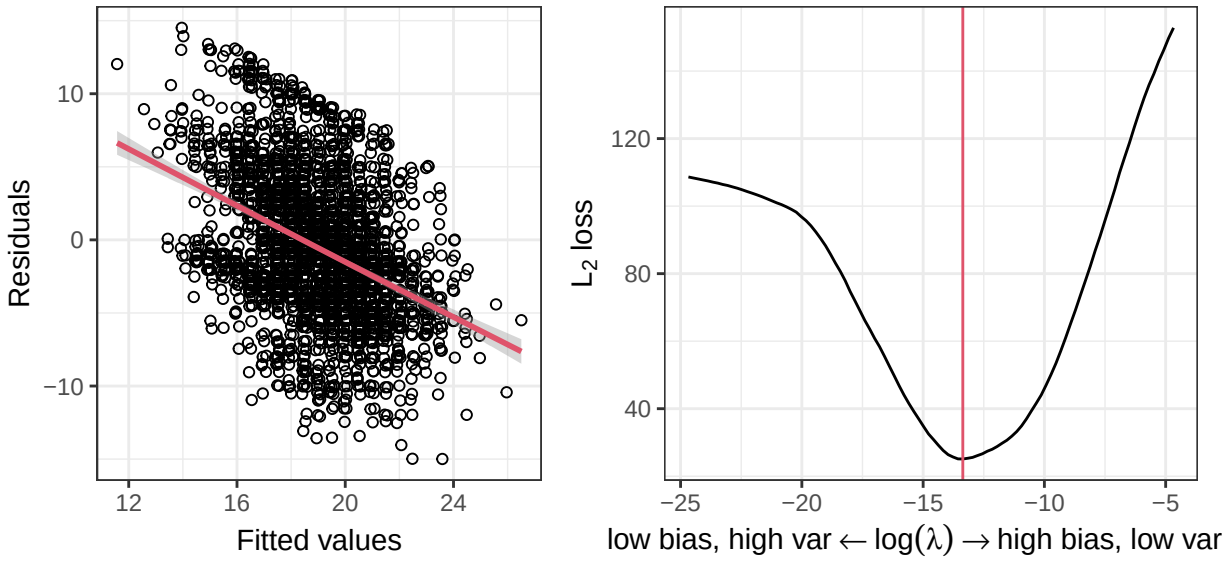


Figure 4: Left: the existence of residual trend, indicates a nonzero bias; right: visual representation of the bias-variance tradeoff via regularisation, the result indicates that the model is more useful than null model.

Notwithstanding the model being shown useful for predicting the rate of breath using the heart rate, it gives biased estimates for predictions. For all *best linear unbiased estimators* (BLUE), such as *least square estimators*, residuals: $\mathbf{r} - \hat{\mathbf{r}}$, are orthogonal (therefore, independent) to the fitted values: $\hat{\mathbf{r}}$. But, the final model gives $\hat{\text{Corr}}(\hat{\mathbf{r}}, \mathbf{r} - \hat{\mathbf{r}}) = -0.3976376$ implying $\text{Bias}(\hat{\mathbf{r}}) = \mathbb{E}(\hat{\mathbf{r}}) - \mathbf{r} \neq \mathbf{0}$. Such bias is introduced intentionally due to regularisation to reduce variance of the estimator $\hat{\mathbf{r}}$. More comparatively:

$$\text{BLUE minimises } \text{Var}(\hat{\mathbf{r}}) \text{ subject to } \text{Bias}(\hat{\mathbf{r}}) = \mathbf{0}$$
$$\text{Regularised estimator minimises } \text{MSE}(\hat{\mathbf{r}}) = \mathbb{E}((\mathbf{r} - \hat{\mathbf{r}})^2) = \text{Var}(\hat{\mathbf{r}}) + \text{Bias}(\hat{\mathbf{r}})^2$$

While BLUE has constraints $\text{Bias}(\hat{\mathbf{r}}) = \mathbf{0}$ (always aims for maximising accuracy first), it unwantedly inflates the variance of the estimator. Whereas regularised estimator appeals to the *bias-variance tradeoff* seeks to directly minimise the cross-validated prediction error (as a function of L_2 loss), that is decomposable to the sum of variance (imprecision) and bias-squared (inaccuracy) for any estimator (Hastie et al., 2009). It follows that, we relaxed the requirement of zero bias via introducing a small bias (sacrificed some accuracy) for a substantially larger drop in variance (massively improved precision) to achieve better predictive performance.

Despite prior anticipation of the bias being nonzero, it is of interest to test for its statistical and practical significance and see if it is acceptably low. For heart-rate-derived breath rate: $\hat{\mathbf{r}}$, as an estimator for breath rate based on chest movements: $\mathbf{r}$, we can draw a hypothesis test for $\mathbf{r} = \beta\hat{\mathbf{r}}$, and due to the linear nature in projecting the estimator (see Algorithm 4), we have the null and alternative hypotheses:

$$H_0 : \text{The estimator } \hat{\mathbf{r}} \text{ is unbiased } \iff \beta = 1$$
$$H_1 : \text{The estimator } \hat{\mathbf{r}} \text{ is biased } \iff \beta \neq 1$$

Therefore, under H_0 :

$$\begin{aligned}\frac{1}{\sigma^2}(1-\hat{\beta})^\top \hat{\mathbf{r}}^\top \hat{\mathbf{r}}(1-\hat{\beta}) &= \sigma^{-2}\hat{\mathbf{r}}^\top \hat{\mathbf{r}}(1-\hat{\beta})^2 \\ &\sim \chi_1^2 \\ \text{where } \hat{\beta} &= (\hat{\mathbf{r}}^\top \hat{\mathbf{r}})^{-1}\hat{\mathbf{r}}^\top \mathbf{r}\end{aligned}$$

Which gives the Wald's statistic (Wald, 1943):

$$\begin{aligned}W &= \hat{\sigma}^{-2}\hat{\mathbf{r}}^\top \hat{\mathbf{r}}(1-\hat{\beta})^2 \\ &= (\dim(\mathbf{r})-1)||(\mathbf{1}-\hat{\mathbf{r}}(\hat{\mathbf{r}}^\top \hat{\mathbf{r}})^{-1}\hat{\mathbf{r}}^\top)\mathbf{r}||^{-2}\hat{\mathbf{r}}^\top \hat{\mathbf{r}}(1-(\hat{\mathbf{r}}^\top \hat{\mathbf{r}})^{-1}\hat{\mathbf{r}}^\top \mathbf{r})^2 \\ &= (\dim(\mathbf{r})-1)||(\mathbf{1}-\hat{\mathbf{r}}(\hat{\mathbf{r}}^\top \hat{\mathbf{r}})^{-1}\hat{\mathbf{r}}^\top)\mathbf{r}||^{-2}(\hat{\mathbf{r}}^\top \hat{\mathbf{r}}-\hat{\mathbf{r}}^\top \mathbf{r})^2\end{aligned}$$

Applied to the results of our final model, we have the Wald's p-value $\mathbb{P}(\chi_1^2 > W) = 2.983174 \times 10^{-13}$, which provides a very strong evidence for the estimator $\hat{\mathbf{r}}$ being biased. However, a further assessment on the 95% asymptotic confidence interval of β, (0.9521173, 0.9724053), indicates the bias is acceptable.

3 Conclusion and further research

3.1 Model evaluation

Despite we have justified the *goodness of fit* for the model, that the heart rate is useful for predicting respiratory rate, model diagnostics based on the training results might give an optimistic estimate for the prediction error. Hence, to neutrally assess the predictive performance of the model, its predictive power shall be tested in a separate *test* data set (c01-03), via *root mean squared error of prediction* (RMSEP).

When estimated optimal value of the regularisation parameter $\lambda = 1.571851 \times 10^{-6}$ is used in the model and applied to the *test* data set, the RMSEP is approximately 5.49 breaths per minute (or 33.95% off the observed respiratory rate). Unfortunately but not surprisingly, it is difficult to, using cardio-activities alone, both *precisely* and *accurately* predict the respiratory rate using *classical* statistical methods. Nonetheless, the findings in this project delivered compelling evidence for the feasibility of respiratory rate prediction using heart rate as one of the predictors, and a direction for implementation.

A lack of information from the *training* data set is a major reason of the under-satisfactory predictive performance of the model. Despite the records are eight hours long at 100Hz each with approximately 3 million data points, the sufficiency in sample size is only useful for building a model robust to abnormal signal. However, merely five individuals are unlikely able to give an adequate generalisation for the population. As such, preferably, the model should be trained using random sample of at least hundreds (even thousands) of individuals, from target populations (children under 5), as a guidance for further researches in this topic.

3.2 Extending the model

Another major factor responsible for the low predictive power is due to the limitations of the *classical* statistical regime. In the following sections, two possible extensions to the model are briefly discussed.

3.2.1 Bayesian approach and random effects models

The model outlined in this report naively assumed *fixed effects*, such that there exists a single true transformation to map the predictors to fitted values. Contextually, we were claiming that, the relationship of heart rate and respiratory rate is the same for all individuals, and it is unaffected by other factors.

A Bayesian extension to our model into a *random effects model* is able to free us from the assumption of *fixed effects*, which helps us model for additional physiological factors such as weight. It is worth noting that Bayesian statistical learning involves posterior computation, that often relies on stochastic simulation such as Markov chain Monte Carlo, which is much more time-consuming to train than the current model.

3.2.2 Deep learning and double descent

Deep learning with *artificial neural networks* gives another direction for further researches. The advantage of deep learning over *classical* statistics is that it pushes the model beyond *bias-variance tradeoff*. While for *classical* methods the effective number of parameters cannot exceed the training sample size, the model complexity of neural network is not restricted by its training sample. Notwithstanding for both deep learning and *classical* methods, the predictive power suffers whenever the effective parameter size grows toward the training sample size, the over-parameterisation in neural networks allows for a second descent in the loss curve (also referred to as risk curve), in a phenomenon called the *double descent* (Belkin et al., 2019). Such effect of *double descent* has only recently been shown by research in *convolutional neural network*, *residual neural network* and *transformers* (Nakkiran et al., 2019).

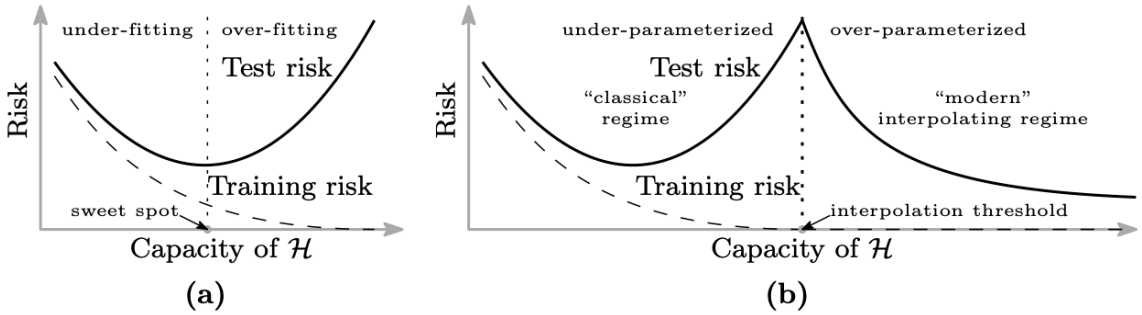


Figure 5: Visualisation in Belkin et al. (2019): (a) Bias-variance tradeoff in classical methods. (b) Double descent loss curve from modern interpolating regime, potentially giving better predictions.

3.3 Open and reproducible research

The source files and data for reproducing the report, the output of the project and history of all edits are available at the GitHub repository szmsu2011/comp90072.

4 Appendix: model training workflow, code and output

4.1 Source code

The source codes for the functions used in the workflow are in the `.R` files at /R.

4.2 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")
```

4.3 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
```

4.4 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
```

4.5 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
```

```
c("RMSEP percentage" = with(resp_dt, sqrt(mean((1 -
  breath_ecg / breath_chest)^2))) * 100)
```

```
#> RMSEP percentage
#>         33.94976
```

References

Becker, R. A., Chambers, J. M., & Wilks, A. R. (1988). *The new s language*. Wadsworth & Brooks/Cole.

Belkin, M., Hsu, D., Ma, S., & Mandal, S. (2019). Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, *116*(32), 15849–15854. https://doi.org/10.1073/pnas.1903070116

Chambers, J. M., & Hastie, T. (1992). *Statistical models in s*. Wadsworth & Brooks/Cole.

Cooley, J. W., & Tukey, J. W. (1965). An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, *19*(90), 297–301. https://doi.org/10.2307/2003354

Giannakopoulos, T., & Pikrakis, A. (2014). *Introduction to audio analysis: A MATLAB approach*. Academic Press. https://doi.org/10.1016/C2012-0-03524-7

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.

Goldberger, A. L., Goldberger, Z. D., & Shvilkin, A. (2017). *Goldberger's clinical electrocardiography: A simplified approach* (9th ed.). Elsevier. https://doi.org/10.1016/C2014-0-03319-9

Green, P. J., & Silverman, B. W. (1994). *Nonparametric regression and generalized linear models: A roughness penalty approach*. Chapman and Hall.

Hastie, T., & Tibshirani, R. (1990). *Generalized additive models*. Chapman and Hall.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed., pp. 37–38). Springer New York. https://hastie.su.domains/Papers/ESLII.pdf

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

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., & Sutskever, I. (2019). *Deep double descent: Where bigger models and more data hurt*. https://arxiv.org/abs/1912.02292

Nyquist, H. (1928). Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers*, *47*(2), 617–644. https://doi.org/10.1109/T-AIEE.1928.5055024

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.

R Core Team. (2023). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

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.

Singleton, R. C. (1979). Mixed radix fast fourier transforms. In IEEE Digital Signal Processing Committee (Ed.), *Programs for digital signal processing*. IEEE Press.

Wald, A. (1943). Tests of statistical hypotheses concerning several parameters when the number of observations is large. *Transactions of the American Mathematical Society*, *54*, 426–482. https://doi.org/10.1090/S0002-9947-1943-0012401-3

World Health Organisation. (2022). *Pneumonia in children*. https://www.who.int/news-room/fact-sheets/detail/pneumonia