

# Data and materials: Rapid recovery of peripheral oxygen saturation in hypoxic COVID-19 patients with ivermectin/doxycycline/zinc multidrug therapy

This document includes the underlying data set and the computer code that was used to conduct the data analysis and generate all of the figures and tables included in our article "Rapid recovery of peripheral oxygen saturation in hypoxic COVID-19 patients with ivermectin/doxycycline/zinc multidrug therapy". The authors of the article are: Sabine Hazan, Adriana C. Vidal, Eleftherios Gkioulekas, Anoja W. Gunaratne, Sibashish Dolai, Robert L. Clancy, Peter A. McCullough, Thomas J. Borody.

## 1. Computer code

The following computer code has been used to generate automatically the tables of the article from the available data.

### 1.1. load-data.r

This file loads the data from the spreadsheet file and calculates number of days for onset of treatment, time to full recovery. It also calculates the SpO<sub>2</sub> deltas. Comorbidity data is then added to the data frame via lists of patient ids for whom the medical charts reported the specific comorbidity. The resulting data frame is filtered to produce a per-protocol subgroup. Then for both intention-to-treat and per-protocol subgroup we calculate the total number of comorbidities per patient and the total number of COVID-19 vulnerable comorbidities per patient.

---

```
#####
# load-data.r
#####

library (dplyr)
library (magrittr)

#####
# Calculate number of days between two dates
date_format <- "%m/%d/%Y"
not.a.date.p <- function (s) s == "NA" | s == "Declined" | s == "Death"
countdays <- function (from, to) ifelse (not.a.date.p (from) | not.a.date.p (to), NA, as.Date (to,
  date_format)-as.Date (from, date_format))

#####
# Convert temperatures from Fahrenheit to Celsius
convert.to.celsius <- function (F) return (round ((5/9)*(F-32), digits=1))

#####
# Load data for intention-to-treat data
# Calculate rx onset time and symptom resolution time
# Calculate the SpO2 deltas
A <- read.csv ("hazan-case-series.csv") %>%
  mutate (temp = convert.to.celsius (as.numeric (temp)),
    basespo2 = as.numeric (basespo2),
    nextdayspo2 = ifelse (is.na (nextdayspo2), NA, as.numeric (nextdayspo2))) %>%
  mutate (rxstartdays = countdays (pospcr, rxstart),
    symptomdays = countdays (rxstart, symptomend),
    negpcrdays = countdays (rxstart, negpcr)) %>%
  mutate (o2delta = nextdayspo2 - basespo2,
    o2stabledelta = nextdayspo2 - 90,
    o2recoverdelta = nextdayspo2 - 95)

#####
# Add comorbidity data to data frame

# Make a vector of 26 patients and set the ones on the list L equal to 1, all others 0
mkitem <- function (L) {
  a <- rep (0, 26)
  a[L] <- 1
  return (a)
}
```

```

}

# Comorbidity data for intention-to-treat group
A <- A %>% mutate (prediabetic = mkitem (c (1, 12, 13, 16, 22)),
  diabetes = mkitem (c (5, 9, 10, 11, 25, 26)),
  pulmembolism = mkitem (c (6)),
  cardiovascular = mkitem (c (1, 2, 10, 21, 22, 24, 25)),
  copd = mkitem (c (2, 9, 25)),
  kidney = mkitem (c (2, 10, 22)),
  immunocompromised = mkitem (c (20)),
  overweight = mkitem (c (4, 12, 14, 24)),
  obese = mkitem (c (1, 13)),
  veryobese = mkitem (c (5, 7, 17, 23)),
  hypertension = mkitem (c (2, 3, 4, 5, 8, 10, 11, 13, 16, 19, 25, 22)),
  apnea = mkitem (c (1, 7, 12, 13, 14, 15, 16, 17, 22, 23)),
  hyperlipidemia = mkitem (c (3, 9, 10, 11, 12, 16, 22, 23, 26)),
  asthma = mkitem (c (8, 14)),
  neurocognitive = mkitem (c (9, 18, 21)),
  thyroid = mkitem (c (5, 6)),
  psychological = mkitem (c (4, 11)),
  rheumatic = mkitem (c (6, 26)),
  gastrointestinal = mkitem (c (3, 4, 10)),
  musculoskeletal = mkitem (c (3, 9, 13)),
  liver = mkitem (c (4)),
  other = mkitem (c (5, 14)))

#####
# For per-protocol data frame
# remove patients 10 and 26
B <- A %>% filter (!(id %in% c (10, 26)))

#####
# Calculate comorbidity counts

# Count the comorbidities
# cmbd = count of all comorbidities
# c19cmbd = count of c19 susceptible comorbidities
add.comorbidities <- function (data) data %>% mutate (cmbd = prediabetic + diabetes + pulmembolism
  + cardiovascular + copd + kidney + immunocompromised + overweight + obese + veryobese +
  hypertension + apnea + hyperlipidemia + asthma + neurocognitive + thyroid + psychological +
  rheumatic + gastrointestinal + musculoskeletal + liver + other) %>%
  mutate (c19cmbd = cmbd - prediabetic - hyperlipidemia - thyroid - rheumatic - gastrointestinal -
    musculoskeletal - other)

# Add cmbd and c19cmbd counts to A and B
A <- A %>% add.comorbidities ()
B <- B %>% add.comorbidities ()

```

---

## 1.2. descriptive.r

Loads the data and produces all of the tables for the descriptive statistics. This includes: the case series table, the demographic data table, the treatment onset and symptom resolution table, the SpO<sub>2</sub> delta table, and the comorbidities table.

```
#####
# descriptive.r
#####

source ("load-data.r")
source ("tables.r")

#####
# Median and IQR for several variables

# report median and IQR
string.median.iqr <- function (x) paste ("median", median (x, na.rm=TRUE), "with IQR", quantile
(x, 1/4, na.rm=TRUE), "to", quantile (x, 3/4, na.rm=TRUE), sep=" ")
report.median.iqr <- function (desc, x) cat (paste (sprintf ("%s:", desc), string.median.iqr (x),
sep=" "), "\n")

# Report on SpO2 deltas
sink ("variables-median-iqr.txt")
cat ("Intention-to-treat measurements: \n")
report.median.iqr ("Patient age", A$age)
report.median.iqr ("Baseline temperature", A$temp)
report.median.iqr ("Baseline SpO2 at room air", A$basespo2)
report.median.iqr ("SpO2 at room air after 24 hours", A$nextdayspo2)
report.median.iqr ("Days before onset of treatment", A$rxstartdays)
report.median.iqr ("Days from onset of treatment to full symptom resolution", A$symptomdays)
report.median.iqr ("SpO2 delta day 1 to day 2", A$o2delta)
report.median.iqr ("SpO2 delta for stabilization over 90% on day 2", A$o2stabledelta)
report.median.iqr ("SpO2 delta for recovery over 95% on day 2", A$o2recoverdelta)
report.median.iqr ("All concurrent comorbidities in patients", A$cmdbd)
report.median.iqr ("Concurrent COVID-19 susceptible comorbidities in patients", A$c19cmdbd)
cat ("\n")
cat ("Per-protocol measurements: \n")
report.median.iqr ("Patient age", B$age)
report.median.iqr ("Baseline temperature", B$temp)
report.median.iqr ("Baseline SpO2 at room air", B$basespo2)
report.median.iqr ("SpO2 at room air after 24 hours", B$nextdayspo2)
report.median.iqr ("Days before onset of treatment", B$rxstartdays)
report.median.iqr ("Days from onset of treatment to full symptom resolution", B$symptomdays)
report.median.iqr ("SpO2 delta day 1 to day 2", B$o2delta)
report.median.iqr ("SpO2 delta for stabilization over 90% on day 2", B$o2stabledelta)
report.median.iqr ("SpO2 delta for recovery over 95% on day 2", B$o2recoverdelta)
report.median.iqr ("All concurrent comorbidities in patients", B$cmdbd)
report.median.iqr ("Concurrent COVID-19 susceptible comorbidities in patients", B$c19cmdbd)
sink ()

#####
# Create case series characteristics table
#####

#####
# Table configuration
table_format_case_series <- "cccccccccccc"
table_header_case_series <- function () list (sprintf ("%s \\\\", paste (textbf ("ID"), textbf
("Age"), textbf ("Race"), textbf ("Sex"), textbf ("Symptoms"), textbf ("Temp"), multicol (2,
paste (textbf ("SpO"), "$\\bm{2}$", sep="")), textbf ("RT-qPCR"), multicol (2, textbf ("Rx
start")), multicol (2, textbf ("Resolved")), sep=" & ")),
paste (cline (7, 8), cline (10, 11), cline (12, 13), sep="")),
sprintf ("&&&&&&& %s && %s \\\\", paste (textbf ("base"), textbf ("date"), textbf ("date"), textbf ("date"), textbf ("date"), textbf ("date"), textbf ("date"), sep=" & ")))
table_caption_case_series <- "Case series subjects, COVID-associated symptoms on presentation, and
```

```

    other characteristics"
table_bottomcaption_case_series <- "\\textbf{NA:} not available;
\\textbf{SOB:} shortness of breath;
\\textbf{PE:} pulmonary embolism;
\\textbf{ID:} identification number;
\\textbf{Age:} patient age in years;
\\textbf{Symptoms:} Patient symptoms upon presentation other than hypoxia;
\\textbf{Temp:} Patient temperature in Celsius upon first presentation;
\\textbf{SpO}_2$\\textbf{:} Room air peripheral oxygen saturation at baseline (base) and
    after 24 hours (+24 h);
\\textbf{RT-qPCR:} Date of first positive RT-qPCR test;
\\textbf{Rx start:} Onset of ivermectin-based multidrug treatment since positive RT-qPCR test
    (days) and date of beginning of IDZCT treatment administration (date);
\\textbf{Resolved:} Days to symptom resolution since initiating treatment (days) and date of
    symptom resolution (date)."
```

```

#####
# Create table
sink ("table-case-series.tex")
use.latex.table (13)
begin.latex.table ("case-series", table_caption_case_series, table_format_case_series,
    table_header_case_series)
for (i in 1:nrow (A)) {
    cat (sprintf ("%s \\\\", paste (A$id[i], A$age[i], A$race[i], A$sex[i], A$symptoms[i],
        A$temp[i], A$basespo2[i], A$nextdayspo2[i], A$pospcr[i], A$rxstartdays[i], A$rxstart[i],
        A$symptomdays[i], A$symptomend[i], sep=" & ")), "\\n")
    if (i<nrow (A)) hline()
}
end.latex.table.with.caption (table_bottomcaption_case_series)
sink ()

#####
# Utilities for counting stuff
#####

#####
# Return percent in string form (n out of N)

percentage.without.the.symbol <- function (n, N) ifelse (n == 0, paste ("$", "0.0", "$", sep=""),
    ifelse (100*(n/N) >= 0.1, paste ("$", sprintf ("%s", round (100*(n/N), digits=1)), "$",
    sep=""), "$< 0.1$"))

#####
# Percent string

percent.string <- function (a) paste (a, "\\ ", "%", sep="")

#####
# LaTeX code for producing able entries for N and %

# Count yes and calculate percentage; save results in dataframe
# data = complete dataset
# filtered_data = filtered dataset for which N and % are needed
count.rows <- function (filtered_data, data) data.frame (all = data %>% nrow (), N = filtered_data
    %>% nrow ()) %>% mutate (percent = percentage.without.the.symbol (N, all))

# Convert the count dataframe to LaTeX code
write.latex <- function (result) sprintf (" $%s$ & %s ", as.character (result$N), as.character
    (result$percent))

#####
# Counters -- return LaTeX code for table entry

```

```

# Count male vs female
# Note: no missing data for intention-to-treat case series
count.male <- function (data) data %>% filter (grepl ("M", sex)) %>% count.rows (data) %>%
  write.latex ()
count.female <- function (data) data %>% filter (grepl ("F", sex)) %>% count.rows (data) %>%
  write.latex ()

# Count an age bracket
# count condition: a <= age <= b
# Note: no missing data for intention-to-treat case series
count.age.bracket <- function (data, a, b) data %>% filter (a <= age, age <= b) %>% count.rows
  (data) %>% write.latex ()

# Count race category
# Note: no missing data for intention-to-treat case series
count.race <- function (data, desc) data %>% filter (grepl (desc, race)) %>% count.rows (data) %>%
  write.latex ()

# Count temperature: a <= temp < b
# Note: no missing data for intention-to-treat case series
count.temp <- function (data, a, b) data %>% filter (a <= temp, temp < b) %>% count.rows (data)
  %>% write.latex ()

# Count days for treatment onset: a <= rxstartdays <= b
# Note no missing data for per-protocol subgroup
count.treatment.onset <- function (data, a, b) data %>% filter (a <= rxstartdays, rxstartdays <=
  b) %>% count.rows (data) %>% write.latex ()

# Count days for symptom resolution: a <= symptomdays <= b
# Note no missing data for per-protocol subgroup
count.symptom.resolution <- function (data, a, b) data %>% filter (a <= symptomdays, symptomdays
  <= b) %>% count.rows (data) %>% write.latex ()

# count entries with a < SpO2 <= b
# note: (0,0) counts missing values
count.spo2.baseline <- function (data, a, b) if (a > 0 & b > 0) data %>% filter (a < basespo2,
  basespo2 <= b) %>% count.rows (data) %>% write.latex () else data %>% filter (is.na
  (basespo2)) %>% count.rows (data) %>% write.latex ()
count.spo2.nextday <- function (data, a, b) if (a > 0 & b > 0) data %>% filter (a < nextdayspo2,
  nextdayspo2 <= b) %>% count.rows (data) %>% write.latex () else data %>% filter (is.na
  (nextdayspo2)) %>% count.rows (data) %>% write.latex ()

# Count some comorbidity
# e.g. A %>% count.something ("hyperlipidemia")
count.something <- function (data, something) data %>% filter (data[[something]] == 1) %>%
  count.rows (data) %>% write.latex ()

# Count patients with n comorbidities
count.cmbd <- function (data, n) data %>% filter (cmbd == n) %>% count.rows (data) %>% write.latex
  ()

# Count patients with n COVID susceptible comorbidities
count.c19cmbd <- function (data, n) data %>% filter (c19cmbd == n) %>% count.rows (data) %>%
  write.latex ()

#####
# Make demographic table
#####

#####
# Functions for writing table rows

add.male <- function (desc) cat (sprintf ("%s \\\\", paste (desc, A %>% count.male (), B %>%
  count.male (), sep=" & ")), "\n")

```

```

add.female <- function (desc) cat (sprintf ("%s \\\\", paste (desc, A %>% count.female (), B %>%
  count.female () , sep=" & ")), "\n")
add.age <- function (desc, a, b) cat (sprintf ("%s \\\\", paste (desc, A %>% count.age.bracket (a,
  b), B %>% count.age.bracket (a, b), sep=" & ")), "\n")
add.race <- function (desc) cat (sprintf ("%s \\\\", paste (desc, A %>% count.race (desc), B %>%
  count.race (desc) , sep=" & ")), "\n")
add.temp.description <- function (a, b, no.fever=FALSE) ifelse (no.fever, "$T < 37$ (no fever)",
  sprintf ("%s \\\leq T < %s$", as.character (a), as.character (b)))
add.temp <- function (a, b, no.fever=FALSE) cat (sprintf ("%s \\\\", paste (add.temp.description
  (a, b, no.fever), A%>% count.temp (a, b), B %>% count.temp (a, b), sep=" & ")), "\n")
spo2.describe <- function (a, b) if (a > 0 & b > 0) paste ("$", percent.string (a), " <
  \\\text{SpO}_2 \\\leq ", percent.string (b), "$", sep="") else paste ("Missing $\\text{SpO}_2$",
  sep="")
add.spo2 <- function (a, b) cat (sprintf ("%s \\\\", paste (spo2.describe (a, b), A %>%
  count.spo2.baseline (a, b), B %>% count.spo2.baseline (a, b), sep=" & ")), "\n")

#####
# Configure demographic table

table_demographic_caption <- "Demographic and clinical characteristics of patients upon
  presentation"
table_demographic_format <- "lcccc"
table_demographic_header <- function () list (sprintf ("%s \\\\", paste (textbf
  ("Characteristic"), multicol (2, textbf ("Intention-to-treat")), multicol (2, textbf
  ("Per-protocol")), sep=" & ")),
  paste (cline (2, 3), cline (4, 5), sep=""),
  sprintf ("%s \\\\", paste ("N", "\\%", "N", "\\%", sep=" & ")))

table_demographic_bottomcaption <- "\\textbf{Intention-to-treat:} Reports on all 26 patients;
\\par\\textbf{Per-protocol:} Reports on 24 patients that adhered to 10-day ivermectin-based
  multidrug treatment;
\\par $\\bm{T}$ Temperature in Celsius prior to commencing ivermectin-based multidrug treatment;
\\par\\textbf{SpO}_2$: Baseline peripheral oxygen saturation on room air prior to
  commencing ivermectin-based multidrug treatment."

#####
# Create demographic table

sink ("table-demographic.tex")
use.latex.table (5)
begin.latex.table ("demographic", table_demographic_caption, table_demographic_format,
  table_demographic_header)
category ("Sex")
add.male ("Male")
add.female ("Female")
hline ()
category ("Age")
add.age ("41 to 50 years", 41, 50)
add.age ("51 to 60 years", 51, 60)
add.age ("61 to 70 years", 61, 70)
add.age ("71 to 80 years", 71, 80)
add.age ("81 to 90 years", 81, 90)
add.age ("91 years or older", 91, 100)
hline ()
category ("Race")
add.race ("Caucasian")
add.race ("Hispanic")
add.race ("Other")
hline ()
category ("Baseline temperature (in Celsius)")
add.temp (0, 37, no.fever=TRUE)
add.temp (37, 38)
add.temp (38, 39)
add.temp (39, 41)
hline ()

```

```

category ("Baseline \\bfsptwo\\ on room air")
add.spo2 (90, 95)
add.spo2 (85, 90)
add.spo2 (80, 85)
add.spo2 (75, 80)
add.spo2 (70, 75)
end.latex.table.with.caption (table_demographic_bottomcaption)
sink ()

#####
# Make treatment onset and symptom resolution table
#####

#####
# Write rows for timing table

add.timing <- function (desc, a, b) cat (sprintf ("%s \\\\", paste (desc, B %>%
count.treatment.onset (a, b), B %>% count.symptom.resolution (a, b), sep=" & ")), "\n")

#####
# Configure the timing table

table_timing_caption <- "Number of days for onset of treatment and symptom resolution for
per-protocol subgroup"
table_timing_format <- "cccc"
table_timing_header <- function () list (sprintf ("%s \\\\", paste (textbf ("Duration"), multicol
(2, textbf ("Rx start")), multicol (2, textbf ("Resolved")), sep=" & ")),
paste (cline (2, 3), cline (4, 5), sep=""),
sprintf ("%s \\\\", paste ("N", "\\%", "N", "\\%", sep=" & ")))

table_timing_bottomcaption <- "\\textbf{Rx start:} Number of days from date of positive RT-qPCR
test to date of start of ivermectin-based treatment.
\\par\\textbf{Resolved:} Number of days from date of start of ivermectin-based treatment to date
of symptom resolution"

#####
# Create the timing table

sink ("table-timing.tex")
use.latex.table (5)
begin.latex.table ("timing", table_timing_caption, table_timing_format, table_timing_header)
add.timing ("0 days", 0, 0)
add.timing ("1 to 5 days", 1, 5)
add.timing ("6 to 10 days", 6, 10)
add.timing ("11 to 20 days", 11, 20)
add.timing ("21 to 30 days", 21, 30)
add.timing ("31 to 40 days", 31, 40)
end.latex.table.with.caption (table_timing_bottomcaption)
sink ()

#####
# Create an SpO2 double-column table including delta
#####

#####
# SpO2 table configuration

# LaTeX code for boldfont SpO2
spo2.bf.string <- paste (textbf ("SpO"), "$\\bm{2}$", sep="")

# Repeat LaTeX code for portion of a row twice
repeat.twice <- function (s) paste (s, s, sep=" & ")

# Table parameters
table_format_spo2 <- strrep ("c", 12)

```

```

table_header_spo2 <- function () list (sprintf ("%s \\\\", repeat.twice (paste (textbf ("ID"),
  multicol (5, spo2.bf.string), sep=" & "))),
  paste (cline (2, 6), cline (8, 12), sep=""),
  sprintf ("%s \\\\", repeat.twice (paste ("", textbf ("day 1"), textbf ("day 2"),
    "$\\bm{\\Delta}$", "$\\bm{\\Delta}_{90}$", "$\\bm{\\Delta}_{95}$", sep=" & "))))
table_caption_spo2 <- "Baseline \\spotwo vs \\spotwo on day 2 at room air for all
  intention-to-treat patients"
table_bottomcaption_spo2 <- "\\textbf{SpO}2\\bm{2}\\textbf{:}$ Peripheral oxygen saturation on
  room air.
  \\par\\textbf{day 1:} Baseline \\spotwo on room air prior to commencing ivermectin-based
    treatment.
  \\par\\textbf{day 2:} Peak \\spotwo on room air by the end of day 2.
  \\par $\\bm{\\Delta}$ = $ Change of \\spotwo on room air from day 1 to day 2.
  \\par $\\bm{\\Delta}_{90}$ = $ Difference between peak \\spotwo on day 2 and the 90\\% \\spotwo
    stabilization threshold.
  \\par $\\bm{\\Delta}_{95}$ = $ Difference between peak \\spotwo on day 2 and the 95\\% \\spotwo
    curative threshold.
  \\par\\textbf{NA:} Not available; \\textbf{ID:} identification number."

#####
# Create the SpO2 table

# For deltas, add a plus sign when positive and return as string
add.plus.sign <- function (n) ifelse (n>0, paste ("+", as.character (n), sep=""), as.character (n))

# Make string with table entries for a specific patient
patient.spo2.before.after <- function (n) paste (A$id[n], A$basespo2[n], A$nextdayspo2[n],
  add.plus.sign (A$o2delta[n]), add.plus.sign (A$o2stabledelta[n]), add.plus.sign
    (A$o2recoveredelta[n]), sep=" & ")

# Create table
sink ("table-spo2-day1-vs-day2.tex")
use.latex.table (12)
begin.latex.table ("spo2-before-after", table_caption_spo2, table_format_spo2, table_header_spo2)
for (i in 1:13) cat (sprintf ("%s & %s \\\\", patient.spo2.before.after (i),
  patient.spo2.before.after (i+13)), "\\n")
end.latex.table.with.caption (table_bottomcaption_spo2)
sink ()

#####
# Comorbidities table
#####

#####
# Functions for writing table rows

# Add a comorbidity count row
# e.g. add.cmbd ("Hyperlipidemia", hyperlipidemia)
add.cmbd <- function (desc, what) {
  something <- deparse (substitute (what))
  cat (sprintf ("%s \\\\", paste (desc, A %>% count.something (something), B %>% count.something
    (something), sep=" & ")), "\\n")
}

# Describe the number of concurrent comorbidites
describe.cmbd.count <- function (n) ifelse (n == 0, "No concurrent comorbidities", ifelse (n == 1,
  "One comorbidity", sprintf ("%s concurrent comorbidities", as.character (n))))

# Add count of patients with n comorbidities
add.cmbd.count <- function (n) cat (sprintf ("%s \\\\", paste (describe.cmbd.count (n), A %>%
  count.cmbd (n), B %>% count.cmbd (n), sep=" & ")), "\\n")

# Add count of patients with n COVID susceptible comorbidities
add.c19cmbd.count <- function (n) cat (sprintf ("%s \\\\", paste (describe.cmbd.count (n), A %>%
  count.c19cmbd (n), B %>% count.c19cmbd (n), sep=" & ")), "\\n")

```

```

#####
# Configure the comorbidities table

table_comorbidity_caption <- "Prevalence of comorbidities in patients"
table_comorbidity_format <- "lcccc"
table_comorbidity_header <- function () list (sprintf ("%s \\", paste (textbf ("Comorbidity"),
    multicol (2, textbf ("Intention-to-treat")), multicol (2, textbf ("Per-protocol")), sep=" &
    )),
    paste (cline (2, 3), cline (4, 5), sep=""),
    sprintf ("% & %s \\", paste ("N", "\\%", "N", "\\%", sep=" & ")))
table_comorbidity_bottomcaption <- "\\textbf{Intention-to-treat:} Reports on all 26 patients;
\\textbf{Per-protocol:} Reports on 24 patients that adhered to 10-day treatment;
\\textbf{Other:} includes glaucoma, prostate disease, and essential tremors;
\\textbf{BMI:} Body mass index;
\\textbf{COVID-19:} Coronavirus Disease 2019;
\\textbf{GERD:} Gastroesophageal reflux disease;
\\textbf{HIV/AIDS:} Human immunodeficiency virus, acquired immunodeficiency syndrome;
\\text{\\textbf{kg}}/\\text{\\text{\\textbf{m}}}}\\text{\\textbf{m}}^2$: kilograms per meter squared."

#####
# Make the comorbidities table

sink ("table-comorbidities.tex")
use.latex.table (5)
begin.latex.table ("comorbidities", table_comorbidity_caption, table_comorbidity_format,
    table_comorbidity_header)
category ("COVID-19 susceptible comorbidities")
add.cmbd ("Type 1 or type 2 diabetes", diabetes)
add.cmbd ("Heart or cardiovascular disease", cardiovascular)
add.cmbd ("Chronic obstructive pulmonary disease", copd)
add.cmbd ("Pulmonary embolism", pulmembolism)
add.cmbd ("Kidney disease", kidney)
add.cmbd ("Liver disease (primary biliary cirrhosis)", liver)
add.cmbd ("Immunocompromised state (HIV/AIDS)", immunocompromised)
add.cmbd ("Overweight (BMI: 25.0--29.9 $\\text{kg}/\\text{m}^2$)", overweight)
add.cmbd ("Obese (BMI: 30.0--39.9$\\text{kg}/\\text{m}^2$)", obese)
add.cmbd ("Morbidly obese (BMI: 40 $\\text{kg}/\\text{m}^2$ or more)", veryobese)
add.cmbd ("Hypertension", hypertension)
add.cmbd ("Sleep apnea", apnea)
add.cmbd ("Asthma", asthma)
add.cmbd ("Neurocognitive disorders (dementia or Alzheimer's)", neurocognitive)
add.cmbd ("Psychological disorders (anxiety or depression)", psychological)
hline ()
category ("Other comorbidities")
add.cmbd ("Prediabetic", prediabetic)
add.cmbd ("Hyperlipidemia", hyperlipidemia)
add.cmbd ("Thyroid", thyroid)
add.cmbd ("Rheumatic diseases (gout or Sj\\\\"{o}gren's)", rheumatic)
add.cmbd ("Gastrointestinal disorders (GERD/gastritis)", gastrointestinal)
add.cmbd ("Musculoskeletal disorders (osteoarthritis, osteopathy, or osteoporosis)",
    musculoskeletal)
add.cmbd ("Other", other)
hline ()
category ("Concurrent COVID-19 susceptible comorbidities in patients")
N <- max (A$c19cmbd)
for (n in 0:N) add.c19cmbd.count (n)
hline ()
category ("All concurrent comorbidities in patients")
N <- max (A$cmbd)
for (n in 0:N) add.cmbd.count (n)
end.latex.table.with.caption (table_comorbidity_bottomcaption)
sink ()

```

### 1.3. wilcoxon.r

Runs the Wilcoxon signed-rank test comparing the before and after SpO<sub>2</sub> levels. Also, sensitivity analysis in which missing data is imputed with the baseline SpO<sub>2</sub> levels.

---

```
#####
# wilcoxon.r
#####

source ("load-data.r")
library (rstatix)

#####
# Interface to rstatix routines
# Run the test to compare before and after
wilcoxon.signed.rank.test <- function (before, after) {
  D <- data.frame (A = before, B = after)
  D.long <- D %>% gather (key="group", value="thing", A, B)
  t1 <- D.long %>% wilcox_test (thing~group, paired=TRUE)
  t2 <- D.long %>% wilcox_effsize (thing~group, paired=TRUE, exact=TRUE)
  return (data.frame (N = nrow (D), p.value = t1$p[1], r = t2$effsize[1]) %>%
    mutate (result = sprintf ("N = %i, p = %s, r = %s", N, as.character (p.value), as.character
      (round (r, digits=2)))))
}

# Display results of the test
display.wilcoxon.signed.rank.test <- function (before, after) cat ("Wilcoxon signed rank test
  results:", "\n", sprintf ("%s", wilcoxon.signed.rank.test (before, after)$result), "\n")

#####
# Wilcoxon Signed-Rank test calculations
# Run the comparisons of SpO2 and sensitivity analysis
sink ("wilcoxon-results.txt")

# C is intention-to-treat dataset with missing data removed
C <- A %>% filter (!is.na (nextdayspo2))
cat ("Intention-to-treat comparison with missing data removed \n")
display.wilcoxon.signed.rank.test (C$basespo2, C$nextdayspo2)
cat ("\n")

# C is per-protocol dataset with missing data removed
C <- B %>% filter (!is.na (nextdayspo2))
cat ("Per-protocol comparison with missing data removed \n")
display.wilcoxon.signed.rank.test (C$basespo2, C$nextdayspo2)
cat ("\n")

# C is intention-to-treat dataset with missing data imputed with baseline
# We still remove patient 26 who died before starting treatment
C <- A %>% mutate (imputnextdayspo2 = ifelse (is.na (nextdayspo2), basespo2, nextdayspo2)) %>%
  mutate (nextdayspo2 = imputnextdayspo2)
cat ("Sensitivity analysis \n")
cat ("Intention-to-treat comparison with missing data imputed with baseline \n")
display.wilcoxon.signed.rank.test (C$basespo2, C$nextdayspo2)
cat ("\n")

# C is per-protocol dataset with missing data imputed with baseline
C <- B %>% mutate (imputnextdayspo2 = ifelse (is.na (nextdayspo2), basespo2, nextdayspo2)) %>%
  mutate (nextdayspo2 = imputnextdayspo2)
cat ("Sensitivity analysis \n")
cat ("Per-protocol comparison with missing data imputed with baseline \n")
display.wilcoxon.signed.rank.test (C$basespo2, C$nextdayspo2)
cat ("\n")

sink ()
```

---

#### 1.4. survival.r

Calculates a Kaplan-Meier plot for time to recovery. Also implements Cox regression time to event analysis for predictors for time to full resolution of all symptoms. Univariate regressions considered 41 predictors, one at a time. Based on that, some multivariate Cox regression models were explored. The results are given in Section 3.3.

The set of 41 predictors that we have considered and their respective definitions are given in the following table. For Boolean variables:  $1 \leftarrow \text{true}$  and  $0 \leftarrow \text{false}$ .

Variable	Definition
discontinued.from.trial.p	True if patient was previously enrolled in a hydroxychloroquine-related trial, deemed treatment failure, and his/her participation in the trial was discontinued.
age.above.60.p	True, if patient age $\geq 60$ years.
age	Patient age in years.
sex.is.male.p	True, if patient is male.
race.is.caucasian.p	True, if patient is Caucasian.
race.is.hispanic.p	True, if patient is Hispanic.
spo2.below.85.p	True, if baseline $\text{SpO}_2 \leq 85\%$ on room air.
basespo2	Baseline $\text{SpO}_2$ on room air.
nextdayspo2	$\text{SpO}_2$ on day 2 (after 24 hours) on room air.
o2delta	Change in $\text{SpO}_2$ between baseline and after 24 hours on room air.
late.treatment.p	true if patient started IDZCT treatment within 5 or more days from the onset of symptoms.
rxstartdays	number of days between positive PCR test and beginning of IDZCT treatment.
fever.low.p	True, if temperature $T \geq 37^\circ\text{C}$ at baseline (low grade fever or worse).
fever.moderate.p	True, if temperature $T \geq 38^\circ\text{C}$ at baseline (moderate fever or worse).
fever.high.p	True, if temperature $T \geq 39^\circ\text{C}$ at baseline (high fever or worse).
fever.scale	Ordinal fever scale with $0 \leftarrow (T < 37^\circ\text{C});$ $1 \leftarrow (37^\circ\text{C} \leq T < 38^\circ\text{C});$ $3 \leftarrow (38^\circ\text{C} \leq T < 39^\circ\text{C});$ $4 \leftarrow (39^\circ\text{C} \leq T < 41^\circ\text{C}).$
temp	Temperature in Celsius.
cmbd	Count of all categorized comorbidities.
c19cmbd	Count of all COVID-19 susceptible comorbidities.
diabetes	True, if patient has type 1 or type 2 diabetes.
prediabetic.or.worse.p	True, if patient is either pre-diabetic or has type 1 or type 2 diabetes.
diabetes.scale	Ordinal diabetes scale with $0 \leftarrow$ no diabetes; $1 \leftarrow$ prediabetic; $2 \leftarrow$ diabetic.
pulmembolism	True, if patient had a pulmonary embolism.
cardiovascular	True, if the patient had cardiovascular-related disease.
copd	True, if the patient had COPD.
kidney	True, if the patient had kidney disease.
immunocompromised	True, if the patient was immunocompromised.
overweight.p	True, if the patient was overweight or worse ( $\text{BMI} > 25 \text{ kg/m}^2$ ).
obese.p	True, if the patient was obese or worse ( $\text{BMI} > 30 \text{ kg/m}^2$ ).
veryobese.p	True, if the patient was morbidly obese or worse ( $\text{BMI} > 40 \text{ kg/m}^2$ ).
obesity.scale	Ordinal obesity scale: $0 \leftarrow$ normal weight; $1 \leftarrow$ overweight ( $\text{BMI}$ between 25 to $30 \text{ kg/m}^2$ ); $2 \leftarrow$ obese ( $\text{BMI}$ between 30 to $40 \text{ kg/m}^2$ ); $3 \leftarrow$ morbidly obese ( $\text{BMI} > 40 \text{ kg/m}^2$ ).
hypertension	True, if the patient had hypertension.
apnea	True, if the patient had apnea.
hyperlipidemia	True, if the patient had hyperlipidemia.
asthma	True, if the patient had asthma.
neurocognitive	True, if the patient had neurocognitive disorders (e.g. dementia or Alzheimer's).
thyroid	True, if the patient had thyroid-related disease.
psychological	True, if the patient had psychological disorders (e.g. anxiety or depression).

rheumatic	True, if the patient had a rheumatic disease (e.g. gout or Sjögren).
gastrointestinal)	True, if the patient had a gastrointestinal disorder (e.g. GERD, gastritis).
musculoskeletal	True, if the patient had a musculoskeletal disorder (e.g. osteoarthritis, osteopathy, osteoporosis).

```
#####
# survival.r
#####

source ("load-data.r")
library (survival)

#####
# Make Kaplan-Meier plot on time to recovery

B <- B %>% mutate (symptomdays.status <- c (rep (1, 24))) # No data was censored
km.recovery <- survfit (Surv (B$symptomdays, B$symptomdays.status) ~ 1)
png ("km-recovery-time.png")
plot (km.recovery,
      xlab="Time since treatment onset (days)",
      ylab="Resolution probability",
      main="Full symptoms resolution Kaplan-Meier plot",
      conf.int=TRUE)
abline (v=10, lty=3, lwd=2, col="#BB0000")
dev.off ()

#####
# Create some additional predicates as possible predictors

# discontinued.from.trial.p: participated in clinical trial and was discontinued due to being
#   deemed treatment failure
# age.above.60.p: older greater or equal than 60
# sex.is.male.p: is male
# race.is.caucasian.p: Race is Caucasian
# race.is.hispanic.p: Race is Hispanic
# overweight.p: overweight or worse (BMI > 25)
# obese.p: obese or worse (BMI > 30)
# veryobese.p: morbidly obese (BMI > 40)
# obesity.scale: 0 = normal weight, 1 = overweight, 2 = obese; 3 = morbidly obese
# spo2.below.85.p: baseline SpO2 at room air below 85%
# late.treatment.p: treatment onset greater than or equal to 5 days
# prediabetic.or.worse.p: prediabetic or diabetic
# diabetes.scale: 0 = no diabetes, 1 = prediabetic; 2 = diabetic
# fever.scale: 0 = no fever, 1 = low-grade fever (37 <= T < 38), 2 = moderate grade fever (38 <= T
#   < 39), 3 = high-grade fever (39 <= T)
# fever.low.p: low-grade fever or worse
# fever.moderate.p: moderate fever or worse
# fever.high.p: high-grade fever

B <- B %>% mutate (discontinued.from.trial.p = ifelse (id %in% c (4, 7, 8, 17, 19), 1, 0),
  age.above.60.p = ifelse (age >= 60, 1, 0),
  sex.is.male.p = ifelse (grepl ("M", sex), 1, 0),
  race.is.caucasian.p = ifelse (grepl ("Caucasian", race), 1, 0),
  race.is.hispanic.p = ifelse (grepl ("Hispanic", race), 1, 0),
  obesity.scale = case_when (overweight == 1 ~ 1, obese == 1 ~ 2, veryobese == 1 ~ 3, TRUE ~ 0),
  spo2.below.85.p = ifelse (basespo2 <= 85, 1, 0),
  late.treatment.p = ifelse (rxstartdays >= 5, 1, 0),
  diabetes.scale = case_when (prediabetic == 1 ~ 1, diabetes == 1 ~ 2, TRUE ~ 0),
  fever.scale = case_when ((37 <= temp & temp < 38) ~ 1, (38 <= temp & temp < 39) ~ 2, (39 <=
    temp & temp < 41) ~ 3, TRUE ~ 0)) %>%
mutate (overweight.p = ifelse (obesity.scale >= 1, 1, 0),
  obese.p = ifelse (obesity.scale >= 2, 1, 0),
  veryobese.p = ifelse (obesity.scale >= 3, 1, 0),
  prediabetic.or.worse.p = ifelse (diabetes.scale > 0, 1, 0),
```

```

fever.low.p = ifelse (fever.scale >= 1, 1, 0),
fever.moderate.p = ifelse (fever.scale >= 2, 1, 0),
fever.high.p = ifelse (fever.scale >= 3, 1, 0)

#####
# Exploratory univariate Cox regressions to identify predictors

# Return formula for univariate Cox regression
univariate.coxph.formula <- function (predictor.string, data.string) as.formula (sprintf ("Surv
(%s$symptomdays, %s$symptomdays.status) ~ %s", data.string, data.string, predictor.string))

# Return a data frame describing the results of a Cox regression for a specific predictor
# e.g. coxph.describe.predictor (model, "age", data = B)
# Output data frame includes:
# formula = the formula used as a string
# HR = Hazard Ratio
# (CI.low, CI.high) = 95% confidence interval
# p.value = p value via Wald test
coxph.describe.predictor <- function (model, predictor.string) {

  # Get the summary of model and confirm the predictor exists
  model.summary <- summary (model)$coefficients
  if (!(predictor.string %in% rownames (model.summary))) stop (sprintf ("%s not found in the
  model", predictor.string))

  # Get the row of model.summary containing the information for the specific predictor
  model.summary.predictor <- model.summary[predictor.string, ]

  # Build and return the data frame with all the information
  return (data.frame (predictor = predictor.string,
    beta = model.summary.predictor["coef"],
    std.dev = model.summary.predictor["se(coef)"],
    p.value = model.summary.predictor["Pr(>|z|)"]) %>%
    mutate (HR = exp (beta),
      CI.low = exp (beta-1.96*std.dev),
      CI.high = exp (beta+1.96*std.dev)))
}

# Output a description of the content returned by coxph.describe.predictor in text format
coxph.predictor.report <- function (model.report) cat (sprintf ("%s, HR = %s, 95%% CI: %s -- %s, p
= %s %s", model.report$predictor,
  as.character (round (model.report$HR, digits=2)),
  as.character (round (model.report$CI.low, digits=2)),
  as.character (round (model.report$CI.high, digits=2)),
  as.character (round (model.report$p.value, digits=4)),
  ifelse (model.report$p.value < 0.05, "(**)", "")), "\n")

# Count the number of comparisons
count.coxph.comparisons <- 0

# Do a univariate Cox regression on a specific predictor and report the outcome in text format
check.coxph.predictor <- function (predictor) {
  predictor.string <- deparse (substitute (predictor))
  B.string <- "B"
  count.coxph.comparisons <- 1 + count.coxph.comparisons
  univariate.coxph.formula (predictor.string, B.string) %>% coxph (data=B) %>%
  coxph.describe.predictor (predictor.string) %>% coxph.predictor.report ()
}

# Report on a predictor using whatever model is passed as argument
coxph.predictor <- function (model, predictor) {
  predictor.string <- deparse (substitute (predictor))
  model %>% coxph.describe.predictor (predictor.string) %>% coxph.predictor.report ()
}

```

```

sink ("cox-regressions.txt")
cat ("Multivariate model: fever.scale + cardiovascular + apnea + gastrointestinal", "\n")
adjusted.cox.model <- coxph (Surv (B$symptomdays, B$symptomdays.status) ~ fever.scale +
    cardiovascular + apnea + gastrointestinal, data=B)
coxph.predictor (adjusted.cox.model, fever.scale)
coxph.predictor (adjusted.cox.model, cardiovascular)
coxph.predictor (adjusted.cox.model, apnea)
coxph.predictor (adjusted.cox.model, gastrointestinal)
cat ("\n")
cat ("Test the proportional hazards assumption for a Cox regression model fit \n")
print (cox.zph (adjusted.cox.model))
cat ("\n")
cat ("Run stepwise reduction \n")
reduced.cox.model <- step (adjusted.cox.model)
cat ("\n")
cat ("Reduced model: cardiovascular + apnea + gastrointestinal", "\n")
coxph.predictor (reduced.cox.model, cardiovascular)
coxph.predictor (reduced.cox.model, apnea)
coxph.predictor (reduced.cox.model, gastrointestinal)
cat ("\n")
cat ("Test the proportional hazards assumption for a Cox regression model fit \n")
print (cox.zph (reduced.cox.model))
cat ("\n")
cat ("Multivariate model: cardiovascular + apnea", "\n")
adjusted.cox.model.v1 <- coxph (Surv (B$symptomdays, B$symptomdays.status) ~ cardiovascular +
    apnea, data=B)
coxph.predictor (adjusted.cox.model.v1, cardiovascular)
coxph.predictor (adjusted.cox.model.v1, apnea)
cat ("\n")
cat ("Test the proportional hazards assumption for a Cox regression model fit \n")
print (cox.zph (adjusted.cox.model.v1))
cat ("\n")
cat ("Univariate Cox models", "\n")
check.coxph.predictor (discontinued.from.trial.p)
check.coxph.predictor (age.above.60.p)
check.coxph.predictor (age)
check.coxph.predictor (sex.is.male.p)
check.coxph.predictor (race.is.caucasian.p)
check.coxph.predictor (race.is.hispanic.p)
check.coxph.predictor (spo2.below.85.p)
check.coxph.predictor (basespo2)
check.coxph.predictor (nextdayspo2)
check.coxph.predictor (o2delta)
check.coxph.predictor (late.treatment.p)
check.coxph.predictor (rxstartdays)
check.coxph.predictor (fever.low.p)
check.coxph.predictor (fever.moderate.p)
check.coxph.predictor (fever.high.p)
check.coxph.predictor (fever.scale)
check.coxph.predictor (temp)
check.coxph.predictor (cmbd)
check.coxph.predictor (c19cmbd)
check.coxph.predictor (diabetes)
check.coxph.predictor (prediabetic.or.worse.p)
check.coxph.predictor (diabetes.scale)
check.coxph.predictor (pulmembolism)
check.coxph.predictor (cardiovascular)
check.coxph.predictor (copd)
check.coxph.predictor (kidney)
check.coxph.predictor (immunocompromised)
check.coxph.predictor (overweight.p)
check.coxph.predictor (obese.p)
check.coxph.predictor (veryobese.p)
check.coxph.predictor (obesity.scale)
check.coxph.predictor (hypertension)

```

```
check.coxph.predictor (apnea)
check.coxph.predictor (hyperlipidemia)
check.coxph.predictor (asthma)
check.coxph.predictor (neurocognitive)
check.coxph.predictor (thyroid)
check.coxph.predictor (psychological)
check.coxph.predictor (rheumatic)
check.coxph.predictor (gastrointestinal)
check.coxph.predictor (musculoskeletal)
# liver does not converge (only one data point)
# check.coxph.predictor (liver)
cat ("\n")
cat (sprintf ("Total: %s univariate Cox regressions considered", as.character
(count.coxph.comparisons)), "\n")
sink ()
```

---

## 1.5. linegraph.r

This file loads the data and produces a line graph showing the change in SpO<sub>2</sub> at baseline and after 24 hours for the intention-to-treat group. From the 26 intention-to-treat, data is displayed for all 21 patients for whom there is available data both before and after the intervention.

---

```
source ("load-data.r")

# Select data for 21 patients with available data before and after
D <- A %>% select (id, basespo2, nextdayspo2) %>% filter (!is.na (nextdayspo2))

# Get before and after
before <- D$basespo2
after <- D$nextdayspo2
id <- D$id

# Create the graph
png ("SpO2-linegraph.png")
plot (c (1, 2), c (min (c (before, after)), max (c (before, after))),
      type="n", xaxt = "n", xlab = "Measurements", ylab = "SpO2 on room air",
      main = "Room air SpO2 response to treatment")
axis (1, at=1:2, labels=c ("Baseline", "+24 hours"))

# Line colors
# Patients 10 and 23 received stat 36mg dose, so color red (#BB0000)
# Otherwise use black color (#000000)
choose.line.colors <- function (id) case_when ((id %in% c (10, 23)) ~ "#BB0000", TRUE ~ "#000000")

# Add lines for each patient
N <- length (before)
for (i in 1:N) lines (1:2, c (before[i], after[i]), lwd=1, col = choose.line.colors (id[i]))

# Add the points
points (rep (1, N), before, pch=19)
points (rep (2, N), after, pch=19)

# Add horizontal dotted line at 90% and 95%
abline (h=90, lty=2, lwd=2)
abline (h=95, lty=2, lwd=2)

# Save the file
dev.off ()
```

---

## 1.6. tables.r

Utility for generating L<sup>A</sup>T<sub>E</sub>X code for tables.

---

```
#####
# tables.r
#####

#####
# Make LaTeX tables
#####

# Boldfont used for table headers
textbf <- function (s) paste ("\\textbf{" , s , "}", sep="")

# Multicolumn
multicol <- function (n, description) sprintf ("\\multicolumn{%s}{c}{%s} ", as.character (n),
  description)
multicol.left <- function (n, description) sprintf ("\\multicolumn{%s}{l}{%s} ", as.character (n),
  description)

# Multicolumn category entry
category <- function (s) cat (multicol.left (table_columns, textbf (s)), "\\ \\ \\ \\ \\n")

#####
# Tabularx tables

# Define table lines for tabularx
# N is number of columns
use.latex.table <- function (N) {
  hline <- function () cat ("\\midrule", "\\n")
  bottomrule <- function () cat ("\\bottomrule", "\\n")
  toprule <- function () cat ("\\toprule", "\\n")
  cline <- function (a, b) sprintf ("\\cmidrule(lr){%s-%s}", as.character (a), as.character (b))
  table_columns <- N
}

# The length \\tablewidth is used to control horizontal size of table
# Add \\newlength{\\tablewidth} to preamble of your document
# To load table, one can use a macro like:
# \\newcommand{\\loadtable}[2]{\\setlength{\\tablewidth}{#1\\textwidth}\\input{#1}}

# Open table. Arguments: table label, table caption, table format, table header
begin.latex.table <- function (tablelabel, tablecaption, tableformat, tableheader) {
  the.tableheader <- tableheader ()
  cat ("\\begin{table}[htb!]\\begin{center}", "\\n")
  cat (sprintf ("\\caption{\\label{fig:%s} %s}", tablelabel, tablecaption), "\\n")
  cat ("{\\small", "\\n")
  cat (sprintf ("\\begin{tabularx}{\\tablewidth}{%s}", tableformat), "\\n")
  toprule ()
  for (i in the.tableheader) cat (i, "\\n")
  hline ()
}

# Close table
end.latex.table <- function () {
  bottomrule ()
  cat ("\\end{tabularx}", "\\n")
  cat ("\\end{center}\\end{table}", "\\n")
}

# Close table with bottom caption
# Definition of bottom caption command
# \\newcommand{\\bottomcaption}[3]{\\multicolumn{#1}{l}{\\parbox{#2}{\\vspace{6pt}\\small #3}} \\}
end.latex.table.with.caption <- function (desc) {
  bottomrule ()
```

```

    cat (sprintf ("\\bottomcaption{%i}{\\tablewidth}{%s} \\\\", table_columns, desc), "\\n")
    cat ("\\end{tabularx}", "\\n")
    cat ("}", "\\n")
    cat ("\\end{center}\\end{table}", "\\n")
}

#####
# longtable for doublespaced version of manuscript

# Define table lines for longtable
# N is number of columns
use.latex.longtable <- function (N) {
  hline <- function () cat ("\\hline", "\\n")
  bottomrule <- hline
  toprule <- hline
  cline <- function (a, b) sprintf ("\\cline{%s-%s}", as.character (a), as.character (b))
  table_columns <- N
}

# Open LaTeX longtable for doublespaced versions
begin.latex.longtable <- function (tablelabel, tablecaption, tableformat, tableheader) {
  the.tableheader <- tableheader ()
  cat ("\\begin{center}", "\\n")
  cat ("{\\small", "\\n")
  cat (sprintf ("\\begin{longtable}{%s}", tableformat), "\\n")
  cat (sprintf ("\\caption{\\small %s}", tablecaption), "\\n")
  cat (sprintf ("\\label{fig:%s} \\\\", tablelabel), "\\n")
  toprule ()
  for (i in the.tableheader) cat (i, "\\n")
  hline ()
}

# Close longtable
end.latex.longtable <- function () {
  bottomrule ()
  cat ("\\end{longtable}", "\\n")
  cat ("\\end{center}", "\\n")
}

# Close longtable with bottom caption
end.latex.longtable.with.caption <- function (desc) {
  bottomrule ()
  cat (sprintf ("\\bottomcaption{%i}{\\tablewidth}{%s} \\\\", table_columns, desc), "\\n")
  cat ("\\end{longtable}", "\\n")
  cat ("}", "\\n")
  cat ("\\end{center}", "\\n")
}

```

---

## 1.7. test-tables.tex

Wrapper L<sup>A</sup>T<sub>E</sub>X file to display the tables generated from the data.

---

```
\documentclass[11pt,reqno,fleqn]{article}

% Load packages
\usepackage{geometry}
\geometry{top=0.9in,left=1in, right=1in,bottom=0.9in}
\usepackage{times}
\usepackage{tabularx}
\usepackage{booktabs}
\usepackage{longtable}
\usepackage{lscape}
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{bm}

% Tack notes at the bottom of a table: \bottomcaption{number of columns{width}}{text}
\newcommand{\bottomcaption}[3]{\multicolumn{#1}{l}{\parbox{#2}{\vspace{6pt}\small #3}} \\\}

%=====
% Commands for inserting tables

% This is used to control the width of tabularx tables
\newlength{\tablewidth}

% 8pt and 9pt font setting to use for tables when compiling in 12pt mode
\newcommand{\ninefont}{\fontsize{9}{11}\selectfont}
\newcommand{\eightfont}{\fontsize{8}{10}\selectfont}

% Display table
\newcommand{\loadtable}[2]{\setlength{\tablewidth}{#1\linewidth}\input{#2}}

%=====
% Local macros

\newcommand{\ftemp}[1]{\$#1\;~{\circ}text{F}\$}
\newcommand{\ctemp}[1]{\$#1\;~{\circ}text{C}\$}
\newcommand{\oxygen}[1]{\text{Sp0}_{2}#1\$ }
\newcommand{\spotwo}{\oxygen{}}
\newcommand{\bfspotwo}{\textbf{Sp0}\$bm_{2} \$}

%=====

\begin{document}

\begin{landscape}
{\let\small\eightfont
\renewcommand{\arraystretch}{0.85}
\loadtable{0.92}{table-case-series.tex}}
\end{landscape}

\loadtable{0.58}{table-demographic.tex}
\loadtable{0.40}{table-timing.tex}
\loadtable{0.77}{table-spo2-day1-vs-day2.tex}
\loadtable{1}{table-comorbidities.tex}

\clearpage
\end{document}
```

---

## 2. Data file: hazan-case-series.csv

Case series data file, not including comorbidities.

---

id	age	race	sex	symptoms	temp	basespo2	nextdayspo2	rxstart	symptomend	pospcr	negpcr
1,66		Caucasian	M	"Runny nose, sore throat, dizzy, low energy"	99.3	90	94	12/14/2020			
								12/21/2020	11/6/2020	1/17/2021	
2,62		Caucasian	M	"SOB, chest congestion, productive cough, nausea, vomiting"	105	77	87				
								12/8/2020	12/18/2020	11/30/2020	12/18/2020
3,75		Caucasian	M	"Low energy"	101	88	96	10/26/2020	11/1/2020	10/15/2020	10/30/2020
4,66		Caucasian	F	"Loss of appetite, cough, chills, SOB"	101	97	96	10/26/2020	10/29/2020		
								10/15/2020	10/30/2020		
5,66		Caucasian	F	"Vomiting, weak, body aches, anosmia"	101	89	95	12/18/2020	12/22/2020		
								12/18/2020	NA		
6,43		Caucasian	F	"PE, headache, body ache, cough"	101	88	94	1/26/2021	2/12/2021	1/26/2021	
								2/9/2021			
7,62		Caucasian	M	"Productive cough, headache"	102	86.5	91	11/24/2020	12/8/2020	11/13/2020	
								12/8/2020			
8,57		Caucasian	M	"Cough, nasal congestion, SOB, body aches"	102	88	96	10/27/2020	11/10/2020		
								10/26/2020	11/15/2020		
9,94		Hispanic	F	"Low energy, SOB, confusion, loss of appetite, shaking"	102	88	94	1/10/2021			
								1/20/2021	12/22/2020	NA	
10,66		Hispanic	M	"Cough, SOB, respiratory failure"	100.6	72	87	"Declined"	"Death"		
								12/22/2020	NA		
11,63		Hispanic	F	"Cough, SOB"	102	90	96	1/10/2021	1/20/2021	12/22/2020	NA
12,47		Hispanic	M	"SOB"	104	84	91	12/19/2020	12/25/2020	12/16/2020	NA
13,69		Caucasian	F	"Cough, congestion, rash"	102	88	91	11/17/2020	12/3/2020	11/13/2020	NA
14,69		Caucasian	M	"Post-nasal drip, cough, sinus pain"	98	88	91	11/17/2020	12/3/2020		
								11/13/2020	NA		
15,71		Hispanic	M	"Low energy, productive cough, anosmia"	101	88	NA	12/17/2020	1/5/2021		
								12/13/2020	NA		
16,67		Hispanic	F	"Dry cough, body aches, low energy, anosmia"	100	88	NA	12/17/2020			
								1/5/2021	12/13/2020	NA	
17,46		Caucasian	F	"Diarrhea, rash, renal pain"	102	87	94	8/8/2020	8/19/2020	7/2/2020	NA
18,86		Caucasian	M	"Cough, fever, low energy"	102	88	95	1/9/2021	1/19/2021	1/8/2021	1/19/2021
19,59		Caucasian	F	"Stomach pain, diarrhea, cough, rash"	102	90	95	9/16/2020	9/25/2020		
								8/19/2020	NA		
20,54		Other	M	"Cough, fever, loss of appetite, chills"	101.2	88	NA	10/16/2020	10/28/2020		
								10/15/2020	10/28/2020		
21,92		Caucasian	M	"Low energy"	102	85	91	2/5/2021	2/11/2021	2/2/2021	2/15/2021
22,63		Hispanic	M	"Cough, low energy, loss of appetite"	101.3	90	96	2/2/2021	2/12/2021		
								2/2/2021	NA		
23,57		Hispanic	M	"Cough, SOB"	98	73	90	1/6/2021	2/8/2021	12/30/2020	1/24/2021
24,46		Hispanic	F	"Chest pain, SOB"	98.6	90	NA	2/18/2021	2/24/2021	2/17/2021	NA
25,87		Hispanic	M	"Severe SOB, low energy, trouble walking"	101.6	90	95	2/27/2021	3/5/2021		
								2/17/2021	NA		
26,86		Caucasian	M	"SOB"	102	88	NA	"Declined"	"Death"	10/6/2020	NA

---

### 3. Output files

In addition to L<sup>A</sup>T<sub>E</sub>X tables, the following output text files are generated by the enclosed computer code.

#### 3.1. variables-median-iqr.txt

---

Intention-to-treat measurements:

Patient age: median 66 with IQR 57.5 to 70.5  
Baseline temperature: median 38.6 with IQR 38.3 to 38.9  
Baseline SpO<sub>2</sub> at room air: median 88 with IQR 87.25 to 89.75  
SpO<sub>2</sub> at room air after 24 hours: median 94 with IQR 91 to 95  
Days before onset of treatment: median 4 with IQR 1 to 11  
Days from onset of treatment to full symptom resolution: median 10 with IQR 6 to 14.5  
SpO<sub>2</sub> delta day 1 to day 2: median 6 with IQR 5 to 7  
SpO<sub>2</sub> delta for stabilization over 90% on day 2: median 4 with IQR 1 to 5  
SpO<sub>2</sub> delta for recovery over 95% on day 2: median -1 with IQR -4 to 0  
All concurrent comorbidities in patients: median 4 with IQR 2 to 4  
Concurrent COVID-19 susceptible comorbidities in patients: median 2 with IQR 1.25 to 3

Per-protocol measurements:

Patient age: median 64.5 with IQR 57 to 69.5  
Baseline temperature: median 38.6 with IQR 38.3 to 38.9  
Baseline SpO<sub>2</sub> at room air: median 88 with IQR 87.75 to 90  
SpO<sub>2</sub> at room air after 24 hours: median 94 with IQR 91 to 95.25  
Days before onset of treatment: median 4 with IQR 1 to 11  
Days from onset of treatment to full symptom resolution: median 10 with IQR 6 to 14.5  
SpO<sub>2</sub> delta day 1 to day 2: median 6 with IQR 4.875 to 7  
SpO<sub>2</sub> delta for stabilization over 90% on day 2: median 4 with IQR 1 to 5.25  
SpO<sub>2</sub> delta for recovery over 95% on day 2: median -1 with IQR -4 to 0.25  
All concurrent comorbidities in patients: median 4 with IQR 2 to 4  
Concurrent COVID-19 susceptible comorbidities in patients: median 2 with IQR 1.75 to 3

---

#### 3.2. wilcoxon-results.txt

---

Intention-to-treat comparison with missing data removed

Wilcoxon signed rank test results:

N = 21, p = 7.04e-05, r = 0.87

Per-protocol comparison with missing data removed

Wilcoxon signed rank test results:

N = 20, p = 0.000105, r = 0.87

Sensitivity analysis

Intention-to-treat comparison with missing data imputed with baseline

Wilcoxon signed rank test results:

N = 26, p = 7.04e-05, r = 0.81

Sensitivity analysis

Per-protocol comparison with missing data imputed with baseline

Wilcoxon signed rank test results:

N = 24, p = 0.000105, r = 0.82

---

#### 3.3. cox-regressions.txt

---

Multivariate model: fever.scale + cardiovascular + apnea + gastrointestinal

fever.scale, HR = 1.31, 95% CI: 0.63 -- 2.7, p = 0.47

cardiovascular, HR = 4.65, 95% CI: 1.33 -- 16.21, p = 0.016 (\*\*\*)

apnea, HR = 0.43, 95% CI: 0.15 -- 1.29, p = 0.1317

gastrointestinal, HR = 15.6, 95% CI: 2.4 -- 101.54, p = 0.004 (\*\*\*)

Test the proportional hazards assumption for a Cox regression model fit

	chisq	df	p
fever.scale	0.6894	1	0.41
cardiovascular	0.1084	1	0.74
apnea	0.0182	1	0.89
gastrointestinal	0.6196	1	0.43
GLOBAL	1.6322	4	0.80

Run stepwise reduction

Start: AIC=99.71

Surv(B\$symptomdays, B\$symptomdays.status) ~ fever.scale + cardiovascular +  
apnea + gastrointestinal

	Df	AIC
- fever.scale	1	98.245
<none>		99.706
- apnea	1	100.115
- cardiovascular	1	103.272
- gastrointestinal	1	103.720

Step: AIC=98.24

Surv(B\$symptomdays, B\$symptomdays.status) ~ cardiovascular +  
apnea + gastrointestinal

	Df	AIC
<none>		98.245
- apnea	1	100.832
- gastrointestinal	1	102.450
- cardiovascular	1	103.357

Reduced model: cardiovascular + apnea + gastrointestinal

cardiovascular, HR = 5.41, 95% CI: 1.64 -- 17.84, p = 0.0056 (\*\*\*)

apnea, HR = 0.35, 95% CI: 0.13 -- 0.94, p = 0.0363 (\*\*\*)

gastrointestinal, HR = 16.58, 95% CI: 2.53 -- 108.66, p = 0.0034 (\*\*\*)

Test the proportional hazards assumption for a Cox regression model fit

	chisq	df	p
cardiovascular	0.23617	1	0.63
apnea	0.00805	1	0.93
gastrointestinal	0.63861	1	0.42
GLOBAL	0.80135	3	0.85

Multivariate model: cardiovascular + apnea

cardiovascular, HR = 3.64, 95% CI: 1.2 -- 11.02, p = 0.0223 (\*\*\*)

apnea, HR = 0.32, 95% CI: 0.12 -- 0.83, p = 0.0186 (\*\*\*)

Test the proportional hazards assumption for a Cox regression model fit

	chisq	df	p
cardiovascular	1.317	1	0.25
apnea	0.126	1	0.72
GLOBAL	1.471	2	0.48

Univariate Cox models

discontinued.from.trial.p, HR = 1.37, 95% CI: 0.49 -- 3.86, p = 0.5522

age.above.60.p, HR = 1.45, 95% CI: 0.59 -- 3.54, p = 0.4162

age, HR = 1.02, 95% CI: 0.98 -- 1.05, p = 0.2903

sex.is.male.p, HR = 0.86, 95% CI: 0.38 -- 1.98, p = 0.7263

race.is.caucasian.p, HR = 1.71, 95% CI: 0.68 -- 4.3, p = 0.2513

race.is.hispanic.p, HR = 0.57, 95% CI: 0.22 -- 1.5, p = 0.2519

spo2.below.85.p, HR = 0.72, 95% CI: 0.21 -- 2.49, p = 0.6026

basespo2, HR = 1.2, 95% CI: 0.98 -- 1.46, p = 0.077

nextdayspo2, HR = 1.19, 95% CI: 0.96 -- 1.48, p = 0.1051

o2delta, HR = 0.88, 95% CI: 0.76 -- 1.03, p = 0.1214

late.treatment.p, HR = 1.36, 95% CI: 0.59 -- 3.18, p = 0.4718

rxstartdays, HR = 1.02, 95% CI: 0.98 -- 1.06, p = 0.2708

fever.low.p, HR = 5.57, 95% CI: 0.73 -- 42.33, p = 0.0969

fever.moderate.p, HR = 2.54, 95% CI: 0.82 -- 7.91, p = 0.1065  
fever.high.p, HR = 2.42, 95% CI: 0.54 -- 10.93, p = 0.2509  
fever.scale, HR = 2.04, 95% CI: 1.03 -- 4.01, p = 0.0396 (\*\*\*)  
temp, HR = 1.63, 95% CI: 0.98 -- 2.7, p = 0.0574  
cmbd, HR = 1.14, 95% CI: 0.85 -- 1.54, p = 0.3724  
c19cmbd, HR = 1.45, 95% CI: 0.91 -- 2.32, p = 0.1158  
diabetes, HR = 3, 95% CI: 0.93 -- 9.71, p = 0.0666  
prediabetic.or.worse.p, HR = 1.45, 95% CI: 0.62 -- 3.37, p = 0.3901  
diabetes.scale, HR = 1.54, 95% CI: 0.85 -- 2.78, p = 0.1528  
pulmembolism, HR = 0.48, 95% CI: 0.06 -- 3.66, p = 0.4792  
cardiovascular, HR = 3.85, 95% CI: 1.28 -- 11.65, p = 0.0168 (\*\*\*)  
copd, HR = 2.15, 95% CI: 0.59 -- 7.77, p = 0.2441  
kidney, HR = 1.56, 95% CI: 0.35 -- 7, p = 0.5643  
immunocompromised, HR = 0.94, 95% CI: 0.12 -- 7.17, p = 0.9532  
overweight.p, HR = 0.97, 95% CI: 0.42 -- 2.27, p = 0.9497  
obese.p, HR = 0.62, 95% CI: 0.23 -- 1.69, p = 0.3535  
veryobese.p, HR = 0.52, 95% CI: 0.15 -- 1.78, p = 0.2985  
obesity.scale, HR = 0.88, 95% CI: 0.62 -- 1.25, p = 0.4747  
hypertension, HR = 1.59, 95% CI: 0.7 -- 3.62, p = 0.2723  
apnea, HR = 0.31, 95% CI: 0.12 -- 0.8, p = 0.015 (\*\*\*)  
hyperlipidemia, HR = 0.67, 95% CI: 0.25 -- 1.76, p = 0.4124  
asthma, HR = 0.66, 95% CI: 0.15 -- 2.9, p = 0.5866  
neurocognitive, HR = 2.15, 95% CI: 0.59 -- 7.77, p = 0.2441  
thyroid, HR = 0.98, 95% CI: 0.22 -- 4.33, p = 0.9834  
psychological, HR = 3.1, 95% CI: 0.69 -- 13.96, p = 0.1412  
rheumatic, HR = 0.48, 95% CI: 0.06 -- 3.66, p = 0.4792  
gastrointestinal, HR = 10.71, 95% CI: 1.97 -- 58.31, p = 0.0061 (\*\*\*)  
musculoskeletal, HR = 1.23, 95% CI: 0.36 -- 4.22, p = 0.737

Total: 41 univariate Cox regressions considered

---