# CXR in COVID Analysis

**Dr Aditya Borakati**

*Royal Free Hospital, Pond Street, London, NW3 2QG*
*a.borakati@doctors.org.uk*

**2020-10-06**

/

# 1 Software Environment and Packages

```
R version 4.0.0 (2020-04-24)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19041)
Matrix products: default
locale:
LC_COLLATE=English_United Kingdom.1252  LC_CTYPE=English_United Kingdom.1252
LC_MONETARY=English_United Kingdom.1252 LC_NUMERIC=C
LC_TIME=English_United Kingdom.1252
attached base packages:
stats     graphics  grDevices utils     datasets  methods   base
other attached packages:
corrplot 0.84
  Taiyun Wei and Viliam Simko (2017). R package "corrplot": Visualization of
  a Correlation Matrix (Version 0.84). Available from
  https://github.com/taiyun/corrplot
MKmisc 1.6
  Kohl M (2019). MKmisc: Miscellaneous functions from M. Kohl_. R package
         version 1.6, http://www.stamats.de
epiR 1.0-14
  Mark Stevenson with contributions from Telmo Nunes, Cord Heuer, Jonathon
  Marshall, Javier Sanchez, Ron Thornton, Jeno Reiczigel, Jim Robison-Cox,
  Paola Sebastiani, Peter Solymos, Kazuki Yoshida, Geoff Jones, Sarah
  Pirikahu, Simon Firestone, Ryan Kyle, Johann Popp, Mathew Jay and Charles
  Reynard. (2020). epiR: Tools for the Analysis of Epidemiological Data. R
  package version 1.0-14. https://CRAN.R-project.org/package=epiR
Matching 4.9-7
  Jasjeet S. Sekhon (2011). Multivariate and Propensity Score Matching
  Software with Automated Balance Optimization: The Matching Package for R.
  Journal of Statistical Software, 42(7), 1-52. URL
         http://www.jstatsoft.org/v42/i07/.
MASS 7.3-51.5
  Venables, W. N. & Ripley, B. D. (2002) Modern Applied Statistics with S.
  Fourth Edition. Springer, New York. ISBN 0-387-95457-0
Ordinal 2019.12-10
  Christensen, R. H. B. (2019). ordinal - Regression Models for Ordinal Data. R
         package version    2019.12-10. https://CRAN.R-
         project.org/package=ordinal.
Hmisc 4.4-0
  Frank E Harrell Jr, with contributions from Charles Dupont and many
  others. (2020). Hmisc: Harrell Miscellaneous. R package version 4.4-0.
  https://CRAN.R-project.org/package=Hmisc
Formula 1.2-3
  Achim Zeileis, Yves Croissant (2010). Extended Model Formulas in R:
  Multiple Parts and Multiple Responses. Journal of Statistical Software
  34(1), 1-13. doi:10.18637/jss.v034.i01
lattice 0.20-41
  Sarkar, Deepayan (2008) Lattice: Multivariate Data Visualization with R.
  Springer, New York. ISBN 978-0-387-75968-5
```

```
mice 3.8.0
  Stef van Buuren, Karin Groothuis-Oudshoorn (2011). mice: Multivariate
  Imputation by Chained Equations in R. Journal of Statistical Software,
  45(3), 1-67. URL https://www.jstatsoft.org/v45/i03/.
readxl 1.3.1
  Hadley Wickham and Jennifer Bryan (2019). readxl: Read Excel Files. R
  package version 1.3.1. https://CRAN.R-project.org/package=readxl
finalfit 1.0.1
  Ewen Harrison, Tom Drake and Riinu Ots (2020). finalfit: Quickly Create
  Elegant Regression Results Tables and Plots when Modelling. R package
  version 1.0.1. https://CRAN.R-project.org/package=finalfit
MatchIt 3.0.2
  Daniel E. Ho, Kosuke Imai, Gary King, Elizabeth A. Stuart (2011). MatchIt:
  Nonparametric Preprocessing for Parametric Causal Inference. Journal of
  Statistical Software, Vol. 42, No. 8, pp. 1-28. URL
  http://www.jstatsoft.org/v42/i08/
tableone 0.11.1
  Kazuki Yoshida (2020). tableone: Create 'Table 1' to Describe Baseline
  Characteristics. R package version 0.11.1.
  https://CRAN.R-project.org/package=tableone
forcats 0.5.0
  Hadley Wickham (2020). forcats: Tools for Working with Categorical
  Variables (Factors). R package version 0.5.0.
  https://CRAN.R-project.org/package=forcats
stringr 1.4.0
  Hadley Wickham (2019). stringr: Simple, Consistent Wrappers for Common
  String Operations. R package version 1.4.0.
  https://CRAN.R-project.org/package=stringr
dplyr 0.8.5
  Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2020).
  dplyr: A Grammar of Data Manipulation. R package version 0.8.5.
  https://CRAN.R-project.org/package=dplyr
purrr 0.3.4
  Lionel Henry and Hadley Wickham (2020). purrr: Functional Programming
  Tools. R package version 0.3.4. https://CRAN.R-project.org/package=purrr
readr 1.3.1
  Hadley Wickham, Jim Hester and Romain Francois (2018). readr: Read
  Rectangular Text Data. R package version 1.3.1.
  https://CRAN.R-project.org/package=readr
tidyr 1.0.2
  Hadley Wickham and Lionel Henry (2020). tidyr: Tidy Messy Data. R package
  version 1.0.2. https://CRAN.R-project.org/package=tidyr
tibble 3.0.0
  Hadley Wickham and Lionel Henry (2020). tidyr: Tidy Messy Data. R package
  version 1.0.2. https://CRAN.R-project.org/package=tidyr
ggplot2 3.3.0
  H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag
  New York, 2016.
tidyverse 1.3.0
  Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source
  Software, 4(43), 1686, https://doi.org/10.21105/joss.01686
forestplot 1.9
  Max Gordon and Thomas Lumley (2019). forestplot: Advanced Forest Plot Using
         'grid' Graphics. R package version 1.9.   https://CRAN.R-
         project.org/package=forestplot
MatchThem 0.9.3
  Farhad Pishgar and Noah Greifer (2020). MatchThem: Matching and Weighting
         Multiply Imputed Datasets. R package version 0.9.3. https://CRAN.R-
         project.org/package=MatchThem
```

```
miceadds 3.9-14
 Robitzsch, A., & Grund, S. (2020). miceadds: Some Additional Multiple
         Imputation Functions, Especially for 'mice'. R package version 3.9-14.
         https://CRAN.R-project.org/package=miceadds
cobalt 4.2.2
 Noah Greifer (2020). cobalt: Covariate Balance Tables and Plots. R package
         version 4.2.2.   https://CRAN.R-project.org/package=cobalt
```

## 1.1 Load Packages and Data

### 1.1.1 Load Packages:

```
library(MKmisc)
library(tidyverse)
library(tableone)
library(MatchIt)
library(finalfit)
library(readxl)
library(cobalt)
library(mice)
library(miceadds)
library(Hmisc)
library(epiR)
library(MatchThem)
library(ordinal)
library(forestplot)
```

## 1.2 Power Calculation

**1.2.0.0.0.1 This code calculates the sample size (positive and negative by gold standard test) needed to evaluate a diagnostic test with 56% sensitivity at 80% power with alpha 0.05. The 56% value is the lower confidence reported by Wong et al. and lower sensitivities typically require higher sample sizes, the result is the same whether specificity or sensitivities are passed as arguments, the previously published specificities are higher than sensitivities so for a generous estimate, the sensitivity was used.**

```
power <- power.diagnostic.test(sens = 0.56,
    sig.level = 0.05, delta = 0.1, power = 0.8) %>%
    print()
```

```
 Diagnostic test exact power calculation

           sens = 0.56
              n = 165
             n1 = 165
          delta = 0.1
      sig.level = 0.05
          power = 0.8
           prev = NULL

NOTE: n is number of cases, n1 is number of controls
```

## 2 Load Data:

```
data <- read_csv("FullDataWithCT.csv", col_types = cols(Age = col_integer(),
    Albumin = col_number(), CK = col_number(),
    CT = col_character(), CRP = col_number(),
    DDimer = col_number(), DateOfDeath = col_date(format = "%d/%m/%Y"),
    DateOfDischarge = col_date(format = "%d/%m/%Y"),
    DateOfVisit = col_date(format = "%d/%m/%Y"),
    DateOfSymptomOnset = col_date(format = "%d/%m/%Y"),
    DiastolicBP = col_number(), FiO2 = col_skip(),
    GCS = col_number(), HR = col_number(),
    MRN = col_skip(), NEWS = col_number(),
    `NEWS2(noFiO2)` = col_skip(), Neutrophils = col_number(),
    RR = col_number(), Sats = col_number(),
    `Supplemental Oxygen` = col_skip(), SystolicBP = col_number(),
    Temperature = col_number(), Troponin = col_number(),
    CTBSTI = col_integer()))
```

# 3 Data Cleaning

###### 3.0.0.0.0.1 Format data into factors/ differences between dates:

```
data <- mutate_if(data, is.character, as.factor)
data$DayOfSymptoms <- difftime(data$DateOfVisit,
    data$DateOfSymptomOnset, units = "days")
data$TimeToDeath <- abs(difftime(data$DateOfDeath,
    data$DateOfVisit, units = "days"))
data$DayOfSymptoms <- as.numeric(data$DayOfSymptoms)
data$TimeToDeath <- as.numeric(data$TimeToDeath)
```

### 3.0.0.1 Recode ethnicities as too many options:

###### 3.0.0.1.0.1 This code collapses the ethnicity categories into ‘White’, ‘Black’, ‘South Asian’, ‘Other Asian’, ‘Mixed’ or ‘Other’;

```
data$Ethnicity <- fct_collapse(data$Ethnicity,
    White = c("White - British", "White - Irish",
        "White - Any Other White Background"))
data$Ethnicity <- fct_collapse(data$Ethnicity,
    Black = c("Black - Any Other Black Background",
        "Black or Black British - A0rican",
        "Black or Black British - African",
        "Black or Black British - Caribbean"))
data$Ethnicity <- fct_collapse(data$Ethnicity,
    `South Asian` = c("Asian or Asian British - Bangladeshi",
        "Asian or Asian British - Indian",
        "Asian or Asian British - Pakistani"))
data$Ethnicity <- fct_collapse(data$Ethnicity,
    `Other Asian` = c("Asian - Any Other Asian Background",
        "Other - Chinese"))
data$Ethnicity <- fct_collapse(data$Ethnicity,
    Mixed = c("mixed - Any Other mixed Background",
        "Mixed - Any Other Mixed Background",
        "Mixed - White and Asian", "Mixed - White and Black African",
        "mixed - White and Black Caribbean",
        "Mixed - White and Black Caribbean"))
```

#### 3.0.0.1.0.2 New XR positive column for "Classic Covid" or not:

```
data$XRPositive <- ifelse(data$XRChest ==
    "Classic COVID", "Positive", "Negative")
data$XRPositive <- as.factor(data$XRPositive)
```

## 3.0.1 Follow Up Swabs + Initial Swabs Positive:

#### 3.0.1.0.0.1 Creates new column 'OverallPos' which includes initial RT-PCR swab and follow-up swabs in 30 days of attendance, if any are positive the value will be positive in this column

```
data$OverallPos <- case_when(data$RTPCR ==
    "Positive" | data$FollowUpPos == "Positive" ~
    "Positive")
data$OverallPos <- replace_na(data$OverallPos,
    "Negative")
```

#### 3.0.1.0.0.2 Create new vector with all variable names (i.e. the column headers)

```
explanatory <- names(data)
```

## 3.0.2 Paired XR and RT-PCR data

### 3.0.2.1 Creates new variable 'completedata' which contains only patients who had both CXR and RT-PCR in ED

```
completedata <- filter(data, !is.na(data$XRPositive) &
    !is.na(data$RTPCR))
```

#### 3.0.2.1.1 Remove missing data variable

```
completedata <- completedata[-c(31)]
```

### 3.0.2.2 Format complete data variables

```
completedata$OverallPos <- as.factor(completedata$OverallPos)

completedata$ThirtyDayFU <- as.factor(completedata$ThirtyDayFU)
completedata$TimeToDeath <- abs(difftime(completedata$DateOfDeath,
    completedata$DateOfVisit, units = "days"))

completedata$TimeToDeath <- as.numeric(completedata$TimeToDeath)
```

##### 3.0.2.2.0.1 Set ‘XRChest’ as ordinal variable on scale of ‘Alternative pathology’ as lowest value and ‘Classical COVID’ as highest

```
completedata$XRChest <- ordered(completedata$XRChest,
    levels = c("Alternative pathology", "No abnormalities",
        "Indeterminate", "Classic COVID"))
```

##### 3.0.2.2.0.2 Convert CT BSTI grade column into factor:

```
completedata$CTBSTI <- as.factor(completedata$CTBSTI)
```

# 4 Demographic table of raw data

**4.0.0.0.0.1 This code creates an unformatted demographic table (table 2 in manuscript), for the raw data, stratified by RT-PCR status, significance testing between RT-PCR +ve and -ve groups is carried out automatically using chi squared, t-tests, ANOVA etc.; there is also a column for the proportion of missing data**

```
CreateTableOne(vars = explanatory,
               strata = 'OverallPos',
               data = completedata) -> demogtable

#### List nonnormal factors for summarisation as median / IQR and non
         parametric statistical test

explanatorynnormal<-c("Sats","RR", "GCS", "SystolicBP", "Temperature", "HR",
         "Neutrophils",
+                       "DDimer","Albumin","CRP","CK","Troponin")
as.data.frame(print(demogtable, nonnormal = explanatorynnormal, missing =
         TRUE))->demogtable

write.csv(demogtable, file = "Demogtable.csv")

  Age (mean (SD))                    62.74 (17.72)             66.18 (17.58)
         0.001
  Ethnicity (%)
         0.097
     Other Asian                        29 ( 8.0)                 72 ( 11.8)
     South Asian                        27 ( 7.5)                 38 (  6.2)
     Black                              41 (11.4)                 91 ( 14.9)
     Mixed                               6 ( 1.7)                  6 (  1.0)
     Other - Any Other Ethnic Group     56 (15.5)                105 ( 17.2)
     White                             202 (56.0)                297 ( 48.8)
  Sex = Male (%)                       233 (53.6)                480 ( 62.9)
         0.002
  Sats (median [IQR])                95.00 [92.00, 98.00]      93.00 [88.00,
         96.00]    <0.001 nonnorm
  RR (median [IQR])                  22.00 [20.00, 28.00]      26.00 [20.00,
         32.00]    <0.001 nonnorm
  GCS (median [IQR])                 15.00 [15.00, 15.00]      15.00 [15.00,
         15.00]     0.043 nonnorm
  SystolicBP (median [IQR])         134.00 [119.00, 151.50]   130.00 [115.00,
         145.00]   0.009 nonnorm
  DiastolicBP (mean (SD))            79.54 (16.40)             75.61 (14.51)
         <0.001
  HR (median [IQR])                  96.00 [83.00, 110.00]     94.00 [81.00,
         108.00]    0.092 nonnorm
```

```
Temperature (median [IQR])          37.10 [36.60, 38.00]      37.70 [37.00,
       38.40]    <0.001 nonnorm
XRChest (%)
       <0.001
   Alternative pathology                4 ( 0.9)                 3 (  0.4)
   No abnormalities                   178 (40.9)               136 ( 17.8)
   Indeterminate                       83 (19.1)               169 ( 22.1)
   Classic COVID                      170 (39.1)               455 ( 59.6)
CTPA = PE (%)                          16 (30.2)                28 ( 45.9)
       0.127
Comorbidity = Yes (%)                 297 (79.0)               482 ( 80.3)
       0.669
Dyspnoea = Yes (%)                    274 (69.4)               497 ( 75.5)
       0.034
Neutrophils (median [IQR])           6.42 [4.55, 9.11]         5.25 [3.69,
       7.61]      <0.001 nonnorm
DDimer (median [IQR])             1250.00 [619.00, 3059.00] 1105.00 [626.00,
       2428.50]  0.204 nonnorm
Albumin (median [IQR])              39.00 [35.00, 42.00]      37.00 [34.00,
       40.00]    <0.001 nonnorm
CRP (median [IQR])                  51.00 [13.00, 117.00]     83.00 [42.00,
       158.00]   <0.001 nonnorm
CK (median [IQR])                   91.00 [54.00, 169.00]    146.50 [78.00,
       342.75]   <0.001 nonnorm
Troponin (median [IQR])             19.00 [7.00, 53.00]       20.00 [9.00,
       53.00]      0.278 nonnorm
Admitted = Discharged (%)             104 (24.0)               128 ( 16.8)
       0.003
AdmittedToITU = Yes (%)                 5 ( 1.3)                32 (  4.8)
       0.005
RTPCR = Positive (%)                    0 ( 0.0)               738 ( 96.7)
       <0.001
CT = 1 (%)                             37 (57.8)                26 ( 86.7)
       0.011
NEWS (mean (SD))                     4.36 (3.06)               5.48 (2.71)
       0.032
ThirtyDayFU (%)
       <0.001
   1                                  219 (78.2)               367 ( 58.3)
   2                                   14 ( 5.0)                49 (  7.8)
   3                                   18 ( 6.4)                60 (  9.5)
   4                                   29 (10.4)               154 ( 24.4)
CTBSTI (%)
       <0.001
   0                                   23 (22.1)                 6 (  3.3)
   1                                   52 (50.0)               157 ( 85.8)
   2                                   14 (13.5)                14 (  7.7)
   3                                   15 (14.4)                 6 (  3.3)
DayOfSymptoms (mean (SD))            9.84 (9.63)               8.56 (15.80)
       0.368
TimeToDeath (mean (SD))             50.33 (77.93)             57.76 (70.02)
       0.618
XRPositive = Positive (%)             170 (39.1)               455 ( 59.6)
       <0.001
OverallPos = Positive (%)               0 ( 0.0)               763 (100.0)
```

`

#### 4.0.0.0.0.2 Limited dataset comprising relevant data and those without significant missingness:

```
limcompletedata <- dplyr::select(completedata,
    c("Age", "XRChest", "Ethnicity", "Sex",
        "RR", "Sats", "GCS", "Temperature",
        "HR", "SystolicBP", "DiastolicBP",
        "Neutrophils", "DDimer", "CRP", "Troponin",
        "Albumin", "CK", "OverallPos", "Admitted",
        "AdmittedToITU", "ThirtyDayFU", "Dyspnoea",
        "Comorbidity", "XRPositive"))
```

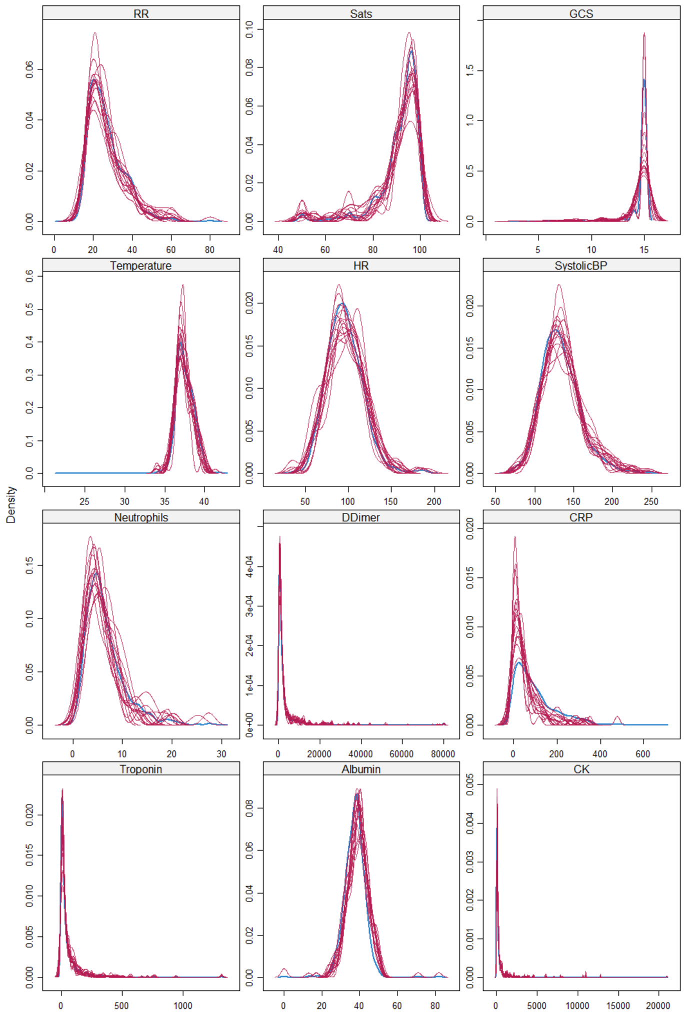
# 5 Imputation

**5.0.0.0.0.1 This code generates 15 imputed datasets using the permuted mean matching method, based on the 'limcompletedata' dataset which has filtered the most relevant fields, with minimal missing data initially**

```
imputed <- mice(limcompletedata, m = 15,
    method = "pmm")
```

**5.0.0.0.0.2 Imputation Diagnostics Density plot, this corresponds to supplementary figure 1:**

```
densityplot(imputed)
```


RR
Sats
GCS
Temperature
HR
SystolicBP
Neutrophils
DDimer
CRP
Troponin
Albumin
CK
Density

# 6 Propensity Score Matching

#### 6.0.0.0.0.1 This code matches data in the imputed datasets on whether the XR was reported classical COVID or not, the matching is done based on the covariates Sex, Age, Comorbidity, Ethnicity and Respiratory Rate

```
library(MatchThem)
#### MatchThem package requires dependent variable to be coded as 0 or 1
imputed[["data"]][["XRPositive"]] %>% recode_factor("Positive" = "1",
         "Negative" = "0") ->imputed[["data"]][["XRPositive"]]
matchthem(
  XRPositive ~ Sex + Age + Comorbidity + Ethnicity + RR,
  data = imputed,
  method = 'nearest',
  verbose = FALSE,
  replace = FALSE,
  ratio = 1,
  caliper = 0.2,
  m.order = "random",) -> matchedtest
### Set XRChest to unordered for binomial analyses
matchedtest[["datasets"]]c(1:15)[["XRChest"]] %>% factor(ordered = FALSE) ->
         matched2[["datasets"]]c(1:15)[["XRChest"]]
```

## 6.1 Match Balance Diagnostics

#### 6.1.0.0.1 Creates plots and table with mean difference and distributation of values in covariates betweeen XR +ve and -ve groups after matching across all imputed datasets:

```
#### Supplementary tables 1,2 and 3:
bal.tab(matchedtest)
#### Supplementary figure 2
bal.plot(matchedtest)
#### Supplementary figure 3:
bal.plot(matchedtest, var.name = "Age", type = "histogram",
    which = "both")
bal.plot(matchedtest, var.name = "Sex", type = "histogram",
    which = "both")
bal.plot(matchedtest, var.name = "Ethnicity",
    type = "histogram", which = "both")
bal.plot(matchedtest, var.name = "RR", type = "histogram",
    which = "both")
bal.plot(matchedtest, var.name = "Comorbidity",
    type = "histogram", which = "both")
#### Supplementary figure 4:
love.plot(matchedtest)
```

# 7 Matched Demographics Table:

#### 7.0.0.0.0.1 Stack matched imputed datasets into one large datset and split into COVID +ve and -ve groups:

```
### 'all=FALSE' gets matched data only
stacked <- MatchThem::complete(matchedtest,
    n = c(1:15), all = FALSE)
stacked <- stacked %>% filter(.imp > 0)
```

#### 7.0.0.0.0.2 Creates demographics table as above, but on propensity matched imputed datasets, corresponds to Table 4:

```
table4 <- CreateTableOne(strata = "OverallPos",
    data = stacked)
#### Means and SD kept as is, mean counts
#### calculated after dividing by 15 (as 15
#### imputed datasets)
```

#### 7.0.0.0.0.3 Creates demographic table stratified by XR Positive or Negative on matched imputed datasets, correpsonds to Table 5:

```
table5 <- CreateTableOne(strata = "XRPositive",
    data = stacked)
#### Means and SD kept as is, mean counts
#### calculated after dividing by 15 (as 15
#### imputed datasets)
```

#### 7.0.0.0.0.4 Summary statistics for pooled data:

```
### Normal means sd
explanatorynorm <- c("Age", "Temperature",
    "HR", "SystolicBP")
summarynormalOverallPos <- stacked %>% group_by(OverallPos) %>%
    summarise_at(vars(explanatorynorm), list(mean.default,
        sd))
summarynormalXRPositive <- stacked %>% group_by(XRPositive) %>%
    summarise_at(vars(explanatorynorm), list(mean.default,
        sd))

### Non normal medians and IQR
summarynnormalOverallPos <- stacked %>% group_by(OverallPos) %>%
    summarise_at(vars(explanatorynnormal),
        list(median, IQR))
summarynnormalXRPositive <- stacked %>% group_by(XRPositive) %>%
    summarise_at(vars(explanatorynnormal),
        list(median, IQR))
```

# 8 Diagnostic Accuracy

#### 8.0.0.1 This section generates the diagnostic accuracy statistics (e.g. sensitivity, specificity) for CXR and CT with RT-PCR as the reference standard using the matched imputed datasets

#### 8.0.0.2 This code creates a contingency table of False/ True Positives and Negatives for Chest X-ray taken from the demographic tables above:

```
contingxr <- matrix(c(305, 243, 125, 187),
    nrow = 2, ncol = 2)

colnames(contingxr) <- c("PCR+", "PCR-")

rownames(contingxr) <- c("XR+", "XR-")
```

##### 8.0.0.2.1 This function calculates diagnostic accuracy test statistics:

```
xraccuracy <- epi.tests(contingxr, conf.level = 0.95)
```

#### 8.0.0.3 Giving the diagnostic accuracy output for CXR in table 3:

```
xraccuracy
          Outcome +    Outcome -      Total
Test +          305          125        430
Test -          243          187        430
Total           548          312        860

Point estimates and 95 % CIs:
---------------------------------------------------------
Apparent prevalence                    0.50 (0.47, 0.53)
True prevalence                        0.64 (0.60, 0.67)
```

```
Sensitivity                              0.56 (0.51, 0.60)
Specificity                              0.60 (0.54, 0.65)
Positive predictive value                0.71 (0.66, 0.75)
Negative predictive value                0.43 (0.39, 0.48)
Positive likelihood ratio                1.39 (1.19, 1.62)
Negative likelihood ratio                0.74 (0.65, 0.84)
----------------------------------------------------------```
```

###### 8.0.0.3.0.1 NB diagnostic accuracy values in table available in list view of xraccuracy variable

---

## 8.1 CT Data and Accuracy

###### 8.1.0.0.0.1 Only those with CT and RT PCR:

```
CTdata <- filter(data, is.na(data$CTBSTI) ==
    FALSE & is.na(data$RTPCR) == FALSE)
```

###### 8.1.0.0.0.2 Select relevant variables

```
CTdata <- dplyr::select(CTdata, c("Age",
    "XRChest", "Ethnicity", "Sex", "RR",
    "Sats", "GCS", "Temperature", "HR", "SystolicBP",
    "DiastolicBP", "Neutrophils", "DDimer",
    "CRP", "Troponin", "OverallPos", "Admitted",
    "AdmittedToITU", "ThirtyDayFU", "Dyspnoea",
    "Comorbidity", "XRPositive", "OverallPos",
    "CTBSTI"))
```

###### 8.1.0.0.0.3 Set RT-PCR as factor:

```
CTdata$OverallPos <- as.factor(CTdata$OverallPos)
```

#### 8.1.0.0.0.4 Rename 1 and 0 to Positive and Negative:

```
CTdata$CTPositive <- ifelse(CTdata$CTBSTI ==
    "1", "Positive", "Negative")
CTdata$CTPositive <- as.factor(CTdata$CTPositive)
```

#### 8.1.0.0.0.5 Regression with CT as outcome variable:

```
CT <- finalfit(
  CTdata,
  "OverallPos",
  c(
    "Age",
    "Sex",
    "RR",
    "GCS",
    "CTPositive",
    "Temperature",
    "HR",
    "SystolicBP",
    "DiastolicBP",
    "Sats",
    "Dyspnoea",
    "Comorbidity"

  ),
  confint_level = 0.95
)
```

#### 8.1.0.0.0.6 Contingency table of True/False Positives and Negatives for CT taken from Regression table:

```
contingct <- matrix(c(CT[7, 4], CT[6, 4],
    CT[7, 3], CT[6, 3]), nrow = 2, ncol = 2)
colnames(contingct) <- c("PCR+", "PCR-")
rownames(contingct) <- c("CT+", "CT-")
contingct <- substr(contingct, start = 1,
    stop = 3)
contingct <- sapply(contingct, as.numeric)
contingct <- matrix(contingct, nrow = 2,
    ncol = 2)
colnames(contingct) <- c("PCR+", "PCR-")

rownames(contingct) <- c("CT+", "CT-")
```

#### 8.1.0.0.0.7 Diagnostic accuracy statistics for CT

```
epi.tests(contingct, conf.level = 0.95) -> ctaccuracy
          Outcome +    Outcome -      Total
Test +          162           55        217
Test -           29           56         85
Total           191          111        302

Point estimates and 95 % CIs:
---------------------------------------------------------
Apparent prevalence                    0.72 (0.66, 0.77)
True prevalence                        0.63 (0.58, 0.69)
Sensitivity                            0.85 (0.79, 0.90)
Specificity                            0.50 (0.41, 0.60)
Positive predictive value              0.75 (0.68, 0.80)
Negative predictive value              0.66 (0.55, 0.76)
Positive likelihood ratio              1.71 (1.41, 2.08)
Negative likelihood ratio              0.30 (0.21, 0.44)
---------------------------------------------------------
```

#### 8.1.0.0.0.8 NB Diagnostic accuracy values found in list view rather than output

## 8.2 CT and XR accuracy comparison

### 8.2.0.1 In this section mean differences of diagnostic accuracy statistics between CT and Chest X-ray with confidence intervals and p-values are calculated

### 8.2.1 Sensitivity

#### 8.2.1.0.0.1 Upper confidence limit for difference in sensitivity

```
ubsens <- (ctaccuracy[["elements"]][["se.up"]] -
    xraccuracy[["elements"]][["se.low"]])
```

#### 8.2.1.0.0.2 Lower confidence limit for difference in sensitivity

```
lbsens <- (ctaccuracy[["elements"]][["se.low"]] -
    xraccuracy[["elements"]][["se.up"]])
```

#### 8.2.1.0.0.3 Mean difference in sensitivity

```
meansens <- ctaccuracy[["elements"]][["se"]] -
    xraccuracy[["elements"]][["se"]]
```

#### 8.2.1.0.0.4 Standard error for sensitivity

```
sesens <- (ubsens - lbsens)/(2 * 1.96)
```

#### 8.2.1.0.0.5 value for difference in sensitivity

```
zsens <- meansens/sesens
```

#### 8.2.1.0.0.6 P-value for difference in sensitivity

```
psens <- exp(-0.717 * zsens - 0.416 * zsens^2)
```

#### 8.2.1.0.0.7 Format values into 'mean difference (95% CI) p-value' rounded to 2 d.p.

```
diffsens <- sprintf("%s (%s-%s)", round(meansens,
    digits = 2), round(lbsens, digits = 2),
    round(ubsens, digits = 2))
diffsensp <- c(diffsens, psens)
```

#### 8.2.1.0.0.8 Subsequent analyses in this section follow the code above

```
## Specificity
ubspec <- (ctaccuracy[["elements"]][["sp.up"]] -
    xraccuracy[["elements"]][["sp.low"]])
lbspec <- (ctaccuracy[["elements"]][["sp.low"]] -
    xraccuracy[["elements"]][["sp.up"]])
meanspec <- ctaccuracy[["elements"]][["sp"]] -
    xraccuracy[["elements"]][["sp"]]
sespec <- (ubspec - lbspec)/(2 * 1.96)
zspec <- meanspec/sespec
pspec <- exp(-0.717 * zspec - 0.416 * zspec^2)
diffspec <- sprintf("%s (%s-%s)", round(meanspec,
    digits = 2), round(lbspec, digits = 2),
    round(ubspec, digits = 2))
diffspecp <- c(diffspec, pspec)


ubda <- (ctaccuracy[["elements"]][["da.up"]] -
    xraccuracy[["elements"]][["da.low"]])
lbda <- (ctaccuracy[["elements"]][["da.low"]] -
    xraccuracy[["elements"]][["da.up"]])
meanda <- ctaccuracy[["elements"]][["da"]] -
    xraccuracy[["elements"]][["da"]]
seda <- (ubda - lbda)/(2 * 1.96)
zda <- meanda/seda
pda <- exp(-0.717 * zda - 0.416 * zda^2)
diffda <- sprintf("%s (%s-%s)", round(meanda,
    digits = 2), round(lbda, digits = 2),
    round(ubda, digits = 2))
diffdap <- c(diffda, pda)
## Positive Likelihood Ratio
ublrpos <- (ctaccuracy[["elements"]][["lrpos.up"]] -
    xraccuracy[["elements"]][["lrpos.low"]])
lblrpos <- (ctaccuracy[["elements"]][["lrpos.low"]] -
    xraccuracy[["elements"]][["lrpos.up"]])
meanlrpos <- ctaccuracy[["elements"]][["lrpos"]] -
    xraccuracy[["elements"]][["lrpos"]]
selrpos <- (ublrpos - lblrpos)/(2 * 1.96)
zlrpos <- meanlrpos/selrpos
plrpos <- exp(-0.717 * zlrpos - 0.416 * zlrpos^2)
difflrpos <- sprintf("%s (%s-%s)", round(meanlrpos,
    digits = 2), round(lblrpos, digits = 2),
    round(ublrpos, digits = 2))
difflrposp <- c(difflrpos, plrpos)
## Negative Likelihood Ratios
ublrneg <- (ctaccuracy[["elements"]][["lrneg.up"]] -
    xraccuracy[["elements"]][["lrneg.low"]])
lblrneg <- (ctaccuracy[["elements"]][["lrneg.low"]] -
    xraccuracy[["elements"]][["lrneg.up"]])
meanlrneg <- ctaccuracy[["elements"]][["lrneg"]] -
    xraccuracy[["elements"]][["lrneg"]]
selrneg <- (ublrneg - lblrneg)/(2 * 1.96)
zlrneg <- meanlrneg/selrneg
plrneg <- exp(-0.717 * zlrneg - 0.416 * zlrneg^2)
difflrneg <- sprintf("%s (%s-%s)", round(meanlrneg,
    digits = 2), round(lblrneg, digits = 2),
    round(ublrneg, digits = 2))
difflrnegp <- c(difflrneg, plrneg)


## Positive Predictive Value
ppv <- (ctaccuracy[["elements"]][["ppv.low"]] -
    xraccuracy[["elements"]][["ppv.up"]])
meanppv <- ctaccuracy[["elements"]][["ppv"]] -
    xraccuracy[["elements"]][["ppv"]]
seppv <- (ubppv - lbppv)/(2 * 1.96)
zppv <- meanppv/seppv
pppv <- exp(-0.717 * zppv - 0.416 * zppv^2)
diffppv <- sprintf("%s (%s-%s)", round(meanppv,
    digits = 2), round(lbppv, digits = 2),
    round(ubppv, digits = 2))
diffppvp <- c(diffppv, pppv)

## Negative Predictive Value
npv <- (ctaccuracy[["elements"]][["npv.low"]] -
    xraccuracy[["elements"]][["npv.up"]])
meannpv <- ctaccuracy[["elements"]][["npv"]] -
    xraccuracy[["elements"]][["npv"]]
senpv <- (ubnpv - lbnpv)/(2 * 1.96)
znpv <- meannpv/senpv
pnpv <- exp(-0.717 * znpv - 0.416 * znpv^2)
diffnpv <- sprintf("%s (%s-%s)", round(meannpv,
    digits = 2), round(lbnpv, digits = 2),
    round(ubnpv, digits = 2))
diffnpvp <- c(diffnpv, pnpv)


## Apparent Prevalence
meantp <- ctaccuracy[["elements"]][["tp"]] -
    xraccuracy[["elements"]][["tp"]]
setp <- (ubtp - lbtp)/(2 * 1.96)
ztp <- meantp/setp
ptp <- exp(-0.717 * ztp - 0.416 * ztp^2)
difftp <- sprintf("%s (%s-%s)", round(meantp,
    digits = 2), round(lbtp, digits = 2),
    round(ubtp, digits = 2))
difftpp <- c(difftp, ptp)

## True Prevalence
meanap <- ctaccuracy[["elements"]][["ap"]] -
    xraccuracy[["elements"]][["ap"]]
```

```
seap <- (ubap - lbap)/(2 * 1.96)
zap <- meanap/seap
pap <- exp(-0.717 * zap - 0.416 * zap^2)
diffap <- sprintf("%s (%s-%s)", round(meanap,
    digits = 2), round(lbap, digits = 2),
    round(ubap, digits = 2))
diffapp <- c(diffap, pap)
```

## 8.3 Intermodality Agreement

#### 8.3.0.0.0.1 This section contains code to analyse the level of agreement in the unmatched CT dataset which contains only data with CT, XR and RT-PCR

#### 8.3.0.0.0.2 First- comparing CT and XR agreement

```
library(irr)
kappa2(c(CTdata$XRPositive, CTdata$CTPositive),
    weight = "squared")
d <- CTdata %>% select(c("CTPositive", "XRPositive"))
View(d)
kappa2(d, weight = "squared")
```

#### 8.3.0.0.0.3 Output:

```
 Cohen's Kappa for 2 Raters (Weights: squared)

 Subjects = 287
   Raters = 2
    Kappa = 0.406

        z = 7.14
  p-value = 9.37e-13
```

#### 8.3.0.0.0.4 The following code compares RT-PCR, CT and XR

```
d2 <- CTdata %>% select(c("CTPositive", "XRPositive",
    "OverallPos"))
View(d2)
kappam.fleiss(d2)
```

#### 8.3.0.0.0.5 Output:

```
 Fleiss' Kappa for m Raters

 Subjects = 287
   Raters = 3
    Kappa = 0.361

        z = 10.6
  p-value = 0
```

## 8.3.1 Diagnostic Accuracy Analysis when Indeterminate Reports of CXR and CT are taken as positive

### 8.3.1.1 XR Indeterminates

#### 8.3.1.1.0.1 New column for positive if indeterminate

```
stacked$XRIndPositive <- ifelse(stacked$XRChest ==
    "Classic COVID" | stacked$XRChest ==
    "Indeterminate", "Positive", "Negative")
stacked$XRIndPositive <- as.factor(stacked$XRIndPositive)
stackedpos <- stacked %>% filter(OverallPos ==
    "Positive")
stackedneg <- stacked %>% filter(OverallPos ==
    "Negative")
summary(stackedpos$XRIndPositive)
summary(stackedneg$XRIndPositive)

contingxrind <- matrix(c(441, 107, 186, 126),
    nrow = 2, ncol = 2)
colnames(contingxrind) <- c("PCR+", "PCR-")

rownames(contingxrind) <- c("XR+", "XR-")
xrindaccuracy <- epi.tests(contingxrind)
```

#### 8.3.1.1.0.2 In this section mean differences of diagnostic accuracy statistics between CT (when CT indeterminates are not counted as positive)and Chest X-ray with confidence intervals and p-values are calculated, follows the same pattern as code previously

```
###### Sensitivity Upper confidence limit for
###### difference in sensitivity

ubsens <- (ctaccuracy[["elements"]][["se.up"]] -
    xrindaccuracy[["elements"]][["se.low"]])
## Lower confidence limit for difference
## in sensitivity
lbsens <- (ctaccuracy[["elements"]][["se.low"]] -
    xrindaccuracy[["elements"]][["se.up"]])
## Mean difference in sensitivity
meansens <- ctaccuracy[["elements"]][["se"]] -
    xrindaccuracy[["elements"]][["se"]]
## Standard error for sensitivity
sesens <- (ubsens - lbsens)/(2 * 1.96)
## Z value for difference in sensitivity
zsens <- meansens/sesens
## P-value for difference in sensitivity
psens <- exp(-0.717 * zsens - 0.416 * zsens^2)
### Format values into 'mean difference
### (95% CI) p-value' rounded to 2 d.p.
diffsens <- sprintf("%s (%s-%s)", round(meansens,
    digits = 2), round(lbsens, digits = 2),
    round(ubsens, digits = 2))
diffsensp <- c(diffsens, psens)

### Subsequent analyses in this section
### follow the code above Specificity
ubspec <- (ctaccuracy[["elements"]][["sp.up"]] -
    xrindaccuracy[["elements"]][["sp.low"]])
lbspec <- (ctaccuracy[["elements"]][["sp.low"]] -
    xrindaccuracy[["elements"]][["sp.up"]])
meanspec <- ctaccuracy[["elements"]][["sp"]] -
    xrindaccuracy[["elements"]][["sp"]]
sespec <- (ubspec - lbspec)/(2 * 1.96)
zspec <- meanspec/sespec
pspec <- exp(-0.717 * zspec - 0.416 * zspec^2)
diffspec <- sprintf("%s (%s-%s)", round(meanspec,
    digits = 2), round(lbspec, digits = 2),
    round(ubspec, digits = 2))
diffspecp <- c(diffspec, pspec)


ubda <- (ctaccuracy[["elements"]][["da.up"]] -
    xrindaccuracy[["elements"]][["da.low"]])
lbda <- (ctaccuracy[["elements"]][["da.low"]] -
    xrindaccuracy[["elements"]][["da.up"]])
meanda <- ctaccuracy[["elements"]][["da"]] -
    xrindaccuracy[["elements"]][["da"]]
seda <- (ubda - lbda)/(2 * 1.96)
```

```
zda <- meanda/seda
pda <- exp(-0.717 * zda - 0.416 * zda^2)
diffda <- sprintf("%s (%s-%s)", round(meanda,
    digits = 2), round(lbda, digits = 2),
    round(ubda, digits = 2))
diffdap <- c(diffda, pda)
## Positive Likelihood Ratio
ublrpos <- (ctaccuracy[["elements"]][["lrpos.up"]] -
    xrindaccuracy[["elements"]][["lrpos.low"]])
lblrpos <- (ctaccuracy[["elements"]][["lrpos.low"]] -
    xrindaccuracy[["elements"]][["lrpos.up"]])
meanlrpos <- ctaccuracy[["elements"]][["lrpos"]] -
    xrindaccuracy[["elements"]][["lrpos"]]
selrpos <- (ublrpos - lblrpos)/(2 * 1.96)
zlrpos <- meanlrpos/selrpos
plrpos <- exp(-0.717 * zlrpos - 0.416 * zlrpos^2)
difflrpos <- sprintf("%s (%s-%s)", round(meanlrpos,
    digits = 2), round(lblrpos, digits = 2),
    round(ublrpos, digits = 2))
difflrposp <- c(difflrpos, plrpos)
## Negative Likelihood Ratios
ublrneg <- (ctaccuracy[["elements"]][["lrneg.up"]] -
    xrindaccuracy[["elements"]][["lrneg.low"]])
lblrneg <- (ctaccuracy[["elements"]][["lrneg.low"]] -
    xrindaccuracy[["elements"]][["lrneg.up"]])
meanlrneg <- ctaccuracy[["elements"]][["lrneg"]] -
    xrindaccuracy[["elements"]][["lrneg"]]
selrneg <- (ublrneg - lblrneg)/(2 * 1.96)
zlrneg <- meanlrneg/selrneg
plrneg <- exp(-0.717 * zlrneg - 0.416 * zlrneg^2)
difflrneg <- sprintf("%s (%s-%s)", round(meanlrneg,
    digits = 2), round(lblrneg, digits = 2),
    round(ublrneg, digits = 2))
difflrnegp <- c(difflrneg, plrneg)


## Positive Predictive Value
ppv <- (ctaccuracy[["elements"]][["ppv.low"]] -
    xrindaccuracy[["elements"]][["ppv.up"]])
meanppv <- ctaccuracy[["elements"]][["ppv"]] -
    xrindaccuracy[["elements"]][["ppv"]]
seppv <- (ubppv - lbppv)/(2 * 1.96)
zppv <- meanppv/seppv
pppv <- exp(-0.717 * zppv - 0.416 * zppv^2)
diffppv <- sprintf("%s (%s-%s)", round(meanppv,
    digits = 2), round(lbppv, digits = 2),
    round(ubppv, digits = 2))
diffppvp <- c(diffppv, pppv)

## Negative Predictive Value
npv <- (ctaccuracy[["elements"]][["npv.low"]] -
    xrindaccuracy[["elements"]][["npv.up"]])
meannpv <- ctaccuracy[["elements"]][["npv"]] -
    xrindaccuracy[["elements"]][["npv"]]
senpv <- (ubnpv - lbnpv)/(2 * 1.96)
znpv <- meannpv/senpv
pnpv <- exp(-0.717 * znpv - 0.416 * znpv^2)
diffnpv <- sprintf("%s (%s-%s)", round(meannpv,
    digits = 2), round(lbnpv, digits = 2),
    round(ubnpv, digits = 2))
diffnpvp <- c(diffnpv, pnpv)


## True Prevalence
meantp <- ctaccuracy[["elements"]][["tp"]] -
    xrindaccuracy[["elements"]][["tp"]]
setp <- (ubtp - lbtp)/(2 * 1.96)
ztp <- meantp/setp
ptp <- exp(-0.717 * ztp - 0.416 * ztp^2)
difftp <- sprintf("%s (%s-%s)", round(meantp,
    digits = 2), round(lbtp, digits = 2),
    round(ubtp, digits = 2))
difftpp <- c(difftp, ptp)

## Apparent Prevalence
meanap <- ctaccuracy[["elements"]][["ap"]] -
    xrindaccuracy[["elements"]][["ap"]]
seap <- (ubap - lbap)/(2 * 1.96)
zap <- meanap/seap
pap <- exp(-0.717 * zap - 0.416 * zap^2)
diffap <- sprintf("%s (%s-%s)", round(meanap,
    digits = 2), round(lbap, digits = 2),
    round(ubap, digits = 2))
diffapp <- c(diffap, pap)
```

## 8.3.1.2 CT Indeterminates

#### 8.3.1.2.0.1 New column for positive if indeterminate

```
CTdata$CTIndPositive <- ifelse(CTdata$CTBSTI ==
    "1" | CTdata$CTBSTI == "2", "Positive",
    "Negative")
CTdata$CTIndPositive <- as.factor(CTdata$CTIndPositive)
valuesctind <- CTdata %>% group_by(OverallPos,
    CTIndPositive) %>% summarise(n = n())
ctcontingind <- matrix(data = c(178, 13,
    70, 41), nrow = 2, ncol = 2)

colnames(ctcontingind) <- c("PCR+ve", "PCR-ve")
rownames(ctcontingind) <- c("CT+ve", "CT-ve")
ctindaccuracy <- epi.tests(ctcontingind)
```

# 9 Pooled Regression after Multiple Imputation and Propensity Score Matching

#### 9.0.0.0.0.1 Binomnal Logistic regression with RT-PCR as dependent variable

```
overallposmatchimp <- matchedtest %>% with(glm(formula(ff_formula(dependent =
          "OverallPos",
    explanatory = c("Age", "Ethnicity", "Sex",
        "RR", "GCS", "Temperature", "HR",
        "SystolicBP", "Neutrophils", "DDimer",
        "CRP", "Troponin", "Albumin", "CK",
        "Sats", "Admitted", "AdmittedToITU",
        "ThirtyDayFUTwo", "Dyspnoea", "Comorbidity",
        "XRChest"))), family = "binomial"),
    all = FALSE)
P <- overallposmatchimp %>% pool()
multivarpooledoverallpos = P %>% fit2df(estimate_name = "OR (multiple
          imputation)",
    exp = TRUE)
```

#### 9.0.0.0.0.2 ‘multivarpooledoverallpos’ produces multivariate odds ratios for each explanatory variable, corresponding to Table 4

## 9.0.1 Pooled Univariate Odds Ratios for OverallPos as dependent variable

#### 9.0.1.0.0.1 This code is run with each of the explanatory variables in table 4 as arguments to produce their respective odds Ratios in table 4

```
overallposmatchimpunivar <- matchedtest %>%
    with(glm(formula(ff_formula(dependent = "OverallPos",
        explanatory = "XRChest")), family = "binomial"))
P <- overallposmatchimpunivar %>% pool()
univarpooledoverallpos = univaroverallpos <- P %>%
    fit2df(estimate_name = "OR (univariate)",
        exp = TRUE)
univaroverallpos
```

### 9.0.2 Binomial Logistic Regression with Positive Chest X-ray Report as Dependent Variable

###### 9.0.2.0.0.1 This code follows the format above to produce univariate and multivariate odds ratios for each explanatory variable for having a positive XR report

### 9.0.3 Univariate XRPositive as dependent

###### 9.0.3.0.0.1 (different explanatory variables passed into function to produce Odds ratios for each)

```
XRChestmatchimp <- matchedtest %>% with(glm(formula(ff_formula(dependent =
         "XRPositive",
    explanatory = "Comorbidity")), family = "binomial"))
P <- XRChestmatchimp %>% pool()
multivarpooledXRChest = univarXRChest <- P %>%
    fit2df(estimate_name = "OR (univariate)",
        exp = TRUE)
univarXRChest
```

### 9.0.4 Multivariate XRPositive as dependent

```
XRChestmatchimp <- matchedtest %>% with(glm(formula(ff_formula(dependent =
         "XRPositive",
    explanatory = c("Age", "OverallPos",
        "Ethnicity", "Sex", "RR", "GCS",
        "Temperature", "HR", "SystolicBP",
        "Neutrophils", "DDimer", "CRP", "Troponin",
        "Albumin", "CK", "Sats", "Admitted",
        "AdmittedToITU", "ThirtyDayFUTwo",
        "Dyspnoea", "Comorbidity"))), family = "binomial"))
P <- XRChestmatchimp %>% pool()
multivarpooledXRChest = multivarXRChest <- P %>%
    fit2df(estimate_name = "OR (multivariate)",
        exp = TRUE)
multivarXRChest
```

## 9.0.5 Pooled Ordinal Logistic Regression with XRPositive as dependent

**9.0.5.0.0.1 This code also produces multivariate odds ratios for table 5, however, uses ordinal linear regression after the CXR report variable is converted to an ordered categorical variable, with alternative pathology as the lowest and classic covid as the highest value (see table 3)**

```
XRChestmatchimpord <- matchedtest %>% with(clm(formula = XRChest ~
    Age + OverallPos + Ethnicity + Sex +
        RR + GCS + Temperature + HR + SystolicBP +
        Neutrophils + DDimer + CRP + Troponin +
        Sats + Admitted + AdmittedToITU +
        ThirtyDayFUTwo + Dyspnoea + Comorbidity))
P <- pool(object = XRChestmatchimpord[["analyses"]])
multivarpooledXRChestord = multivarXRChestord <- P %>%
    fit2df(estimate_name = "OR (multivariate)",
        exp = TRUE)
multivarXRChestord
```

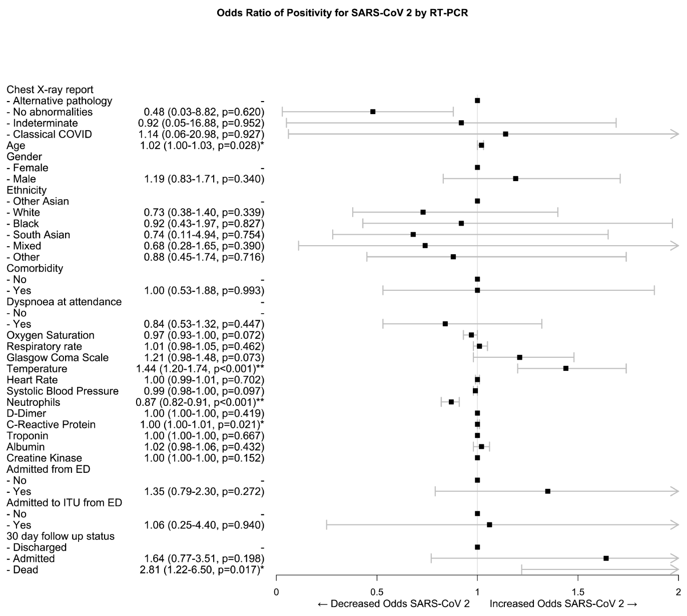
## 9.1 Forest Plots

**9.1.0.0.0.1 Creates forest plots for post matched regression tables above:**

```
Figure1Forest <- read_excel("Figure1Forest.xlsx",
    col_types = c("text", "numeric", "numeric",
        "numeric", "text", "text"))

tabletext1 <- cbind(Figure1Forest$explanatory,
    Figure1Forest$summary)
forestplot(tabletext1, Figure1Forest$Mean,
    Figure1Forest$Lower, Figure1Forest$Upper,
    is.summary = FALSE, clip = c(0, 2), xlab = "<U+2190> Decreased Odds SARS-
        CoV 2       Increased Odds SARS-CoV 2 <U+2192>",
    zero = 1, cex = 0.9, lineheight = unit(6,
        "mm"), boxsize = 0.4, colgap = unit(6,
        "mm"), lwd.ci = 2, ci.vertices = TRUE,
    ci.vertices.height = 0.4, title = "Odds Ratio of Positivity for SARS-CoV 2
        by RT-PCR",
    txt_gp = fpTxtGp(label = gpar(cex = 1.25),
        ticks = gpar(cex = 1.1), xlab = gpar(cex = 1.2),
        title = gpar(cex = 1.2)), graphwidth = unit(200,
        "mm"))
```

### 9.1.0.0.0.2 Figure 2:



| | |
|---|---|
| Chest X-ray report | |
| - Alternative pathology | - |
| - No abnormalities | 0.48 (0.03-8.82, p=0.620) |
| - Indeterminate | 0.92 (0.05-16.88, p=0.952) |
| - Classical COVID | 1.14 (0.06-20.98, p=0.927) |
| Age | 1.02 (1.00-1.03, p=0.028)* |
| Gender | |
| - Female | - |
| - Male | 1.19 (0.83-1.71, p=0.340) |
| Ethnicity | |
| - Other Asian | - |
| - White | 0.73 (0.38-1.40, p=0.339) |
| - Black | 0.92 (0.43-1.97, p=0.827) |
| - South Asian | 0.74 (0.11-4.94, p=0.754) |
| - Mixed | 0.68 (0.28-1.65, p=0.390) |
| - Other | 0.88 (0.45-1.74, p=0.716) |
| Comorbidity | |
| - No | - |
| - Yes | 1.00 (0.53-1.88, p=0.993) |
| Dyspnoea at attendance | - |
| - No | - |
| - Yes | 0.84 (0.53-1.32, p=0.447) |
| Oxygen Saturation | 0.97 (0.93-1.00, p=0.072) |
| Respiratory rate | 1.01 (0.98-1.05, p=0.462) |
| Glasgow Coma Scale | 1.21 (0.98-1.48, p=0.073) |
| Temperature | 1.44 (1.20-1.74, p<0.001)** |
| Heart Rate | 1.00 (0.99-1.01, p=0.702) |
| Systolic Blood Pressure | 0.99 (0.98-1.00, p=0.097) |
| Neutrophils | 0.87 (0.82-0.91, p<0.001)** |
| D-Dimer | 1.00 (1.00-1.00, p=0.419) |
| C-Reactive Protein | 1.00 (1.00-1.01, p=0.021)* |
| Troponin | 1.00 (1.00-1.00, p=0.667) |
| Albumin | 1.02 (0.98-1.06, p=0.432) |
| Creatine Kinase | 1.00 (1.00-1.00, p=0.152) |
| Admitted from ED | |
| - No | - |
| - Yes | 1.35 (0.79-2.30, p=0.272) |
| Admitted to ITU from ED | |
| - No | - |
| - Yes | 1.06 (0.25-4.40, p=0.940) |
| 30 day follow up status | |
| - Discharged | - |
| - Admitted | 1.64 (0.77-3.51, p=0.198) |
| - Dead | 2.81 (1.22-6.50, p=0.017)* |

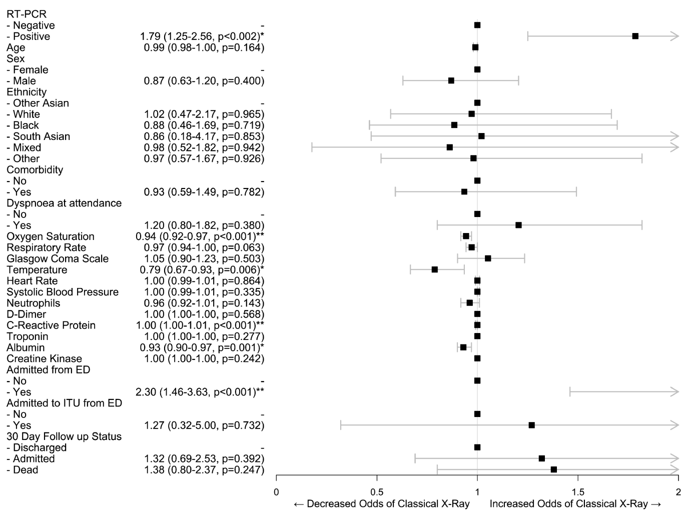
### 9.1.0.0.0.3 Figure 3 (XR dependent):

```
Figure2Forest <- read_excel("Figure2Forest.xlsx",
                           col_types = c("text", "numeric", "numeric",
                                         "numeric", "text", "text"))

tabletext2<-cbind(Figure2Forest$explanatory,Figure2Forest$summary)
forestplot (tabletext2, Figure2Forest$Mean,
            Figure2Forest$Lower, Figure2Forest$Upper, is.summary = FALSE,
            clip = c(0, 2),
          xlab="\u2190 Decreased Odds of Classical X-Ray       Increased Odds
        of Classical X-Ray \u2192",
          zero=1, cex=0.9, lineheight = unit(6,"mm"), boxsize=0.5,
        colgap=unit(6,"mm"),
          lwd.ci=2, ci.vertices=TRUE, ci.vertices.height = 0.4,
          title="Odds Ratio of Classical COVID-19 Findings on Chest X-Ray",
          txt_gp=fpTxtGp(label=gpar(cex=1.25),
                         ticks=gpar(cex=1.1),
                         xlab=gpar(cex = 1.2),
                         title=gpar(cex = 1.2)),
          graphwidth = unit(200,"mm")

)
```

Odds Ratio of Classical COVID-19 Findings on Chest X-Ray



| | |
|---|---|
| RT-PCR | |
| - Negative | - |
| - Positive | 1.79 (1.25-2.56, p<0.002)* |
| Age | 0.99 (0.98-1.00, p=0.164) |
| Sex | |
| - Female | - |
| - Male | 0.87 (0.63-1.20, p=0.400) |
| Ethnicity | |
| - Other Asian | - |
| - White | 1.02 (0.47-2.17, p=0.965) |
| - Black | 0.88 (0.46-1.69, p=0.719) |
| - South Asian | 0.86 (0.18-4.17, p=0.853) |
| - Mixed | 0.98 (0.52-1.82, p=0.942) |
| - Other | 0.97 (0.57-1.67, p=0.926) |
| Comorbidity | |
| - No | - |
| - Yes | 0.93 (0.59-1.49, p=0.782) |
| Dyspnoea at attendance | |
| - No | - |
| - Yes | 1.20 (0.80-1.82, p=0.380) |
| Oxygen Saturation | 0.94 (0.92-0.97, p<0.001)** |
| Respiratory Rate | 0.97 (0.94-1.00, p=0.063) |
| Glasgow Coma Scale | 1.05 (0.90-1.23, p=0.503) |
| Temperature | 0.79 (0.67-0.93, p=0.006)* |
| Heart Rate | 1.00 (0.99-1.01, p=0.864) |
| Systolic Blood Pressure | 1.00 (0.99-1.01, p=0.335) |
| Neutrophils | 0.96 (0.92-1.01, p=0.143) |
| D-Dimer | 1.00 (1.00-1.00, p=0.568) |
| C-Reactive Protein | 1.00 (1.00-1.01, p<0.001)** |
| Troponin | 1.00 (1.00-1.00, p=0.277) |
| Albumin | 0.93 (0.90-0.97, p=0.001)* |
| Creatine Kinase | 1.00 (1.00-1.00, p=0.242) |
| Admitted from ED | |
| - No | - |
| - Yes | 2.30 (1.46-3.63, p<0.001)** |
| Admitted to ITU from ED | |
| - No | - |
| - Yes | 1.27 (0.32-5.00, p=0.732) |
| 30 Day Follow up Status | |
| - Discharged | - |
| - Admitted | 1.32 (0.69-2.53, p=0.392) |
| - Dead | 1.38 (0.80-2.37, p=0.247) |

## 9.2 Correlation Matrix

#### 9.2.0.0.0.1 This section creates a plot of correlation between all the variables in the raw data

```
library(corrplot)
library(Hmisc)
```

#### 9.2.0.0.0.2 Relevel factors so relevant value is first

```
data$XRPositive <- relevel(data$XRPositive,
    "Negative")

data$Admitted <- relevel(data$Admitted, "Discharged")
data$AdmittedToITU <- relevel(data$AdmittedToITU,
    "No")
```

#### 9.2.0.0.0.3 New variable for correlation matrix

```
cor <- data
```

#### 9.2.0.0.0.4 Remove variables with high missings/ data which won’t work e.g. date, RT-PCR removed as it only represents initial ED swab, OverallPos used instead as this includes susequent swabs in 30 days

```
cor<-subset(data, select = -c(CT,DateOfDeath,DateOfDischarge,RTPCR,

         DateOfVisit,DateOfSymptomOnset,FollowUpPos,TimeToDeath,NEWS))'
```

#### 9.2.0.0.0.5 Format and re-name values

```
cor$CTPositive <- ifelse(cor$CTBSTI == "1",
    "Positive", "Negative")
cor$CTPositive <- as.factor(cor$CTPositive)
cor$CTPositive <- relevel(cor$CTPositive,
    "Negative")
cor$Death <- as.factor(ifelse(cor$ThirtyDayFU ==
    "4", "Dead", "Alive"))
cor$Death <- relevel(cor$Death, "Alive")
cor$OverallPos <- as.factor(cor$OverallPos)
cor <- sapply(cor, as.numeric)
```

#### 9.2.0.0.0.6 Create new numerical correlation matrix

```
cormatrixall <- cor(cor, method = "spearman",
    use = "pairwise.complete.obs")
```

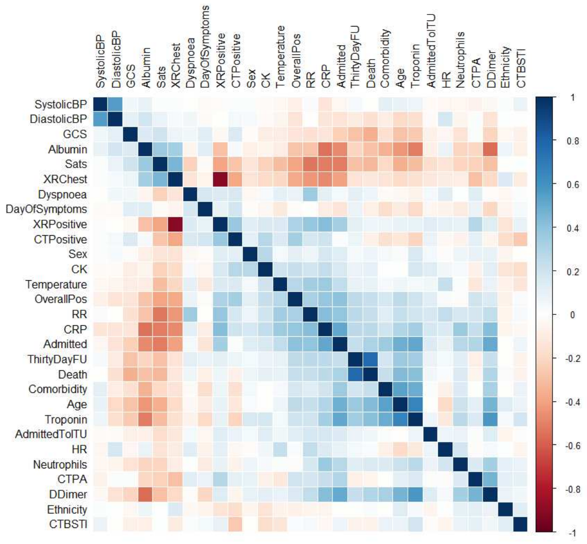
#### 9.2.0.0.0.7 This variable also contains p-values so identification of only significant correlations is possible:

```
cormatrixall2 <- rcorr(as.matrix(cor), type = "spearman")
```

#### 9.2.0.0.0.8 Function to create and format correlation matrix plot

```
corrplot(cormatrixall2$r, method = "color",
    type = "full", order = "hclust", p.mat = cormatrixall2$p,
    sig.level = 0.05, insig = "blank", tl.col = "black",
    outline = "white", title = "Correlation Matrix of Explanatory and Outcome
        Variables",
    line = -1, cex.main = 2, adj.main = 0.5)
```

**Correlation Matrix of Explanatory and Outcome Variables**



## 9.3 STARD Flow Diagram

**9.3.0.0.0.1 See instructions from https://www.r-bloggers.com/flow-charts-in-r/**

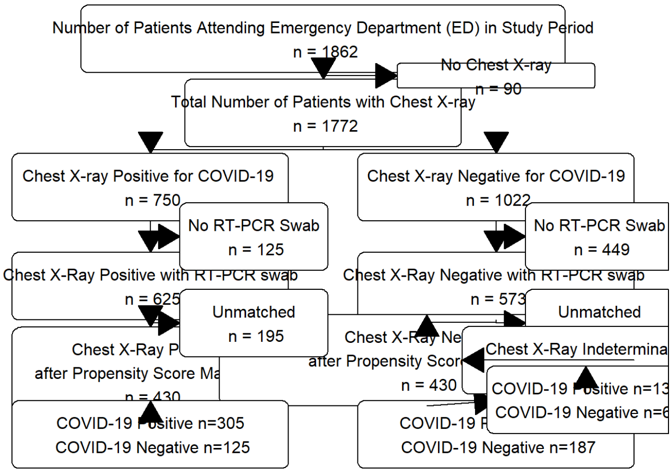
**9.3.0.0.0.2 Produces flow charts in Figure 1, (images need to be stretched out, output as svgs)**

```
library(grid)
library(Gmisc)

grid.newpage()
# set some parameters to use repeatedly
leftx <- 0.25
```

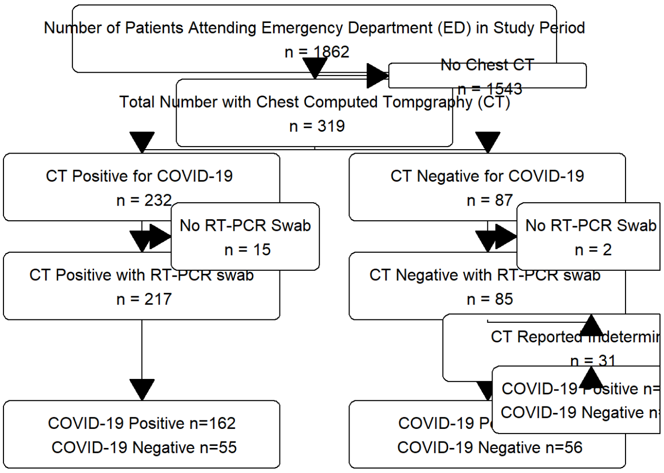
```r
midx <- 0.5
rightx <- 0.75
width <- 0.4
gp <- gpar(fill = "white")
# create boxes
(totalattendance <- boxGrob("Number of Patients Attending Emergency Department
         (ED) in Study Period\n n = 1862",
    x = midx, y = 0.9, box_gp = gp, width = 0.7))

(numberwithxr <- boxGrob("Total Number of Patients with Chest X-ray\n n =
         1772",
    x = midx, y = 0.75, box_gp = gp, width = width))
# connect boxes like this
connectGrob(totalattendance, numberwithxr,
    "v")

(numberwithoutxr <- boxGrob("No Chest X-ray\n n = 90",
    x = rightx, y = 0.825, box_gp = gp, width = unit(2,
        "inch"), height = 0.05))

connectGrob(totalattendance, numberwithoutxr,
    "-")

(XRPos <- boxGrob("Chest X-ray Positive for COVID-19 \n n = 750",
    x = leftx, y = 0.6, box_gp = gp, width = width))
(XRNeg <- boxGrob("Chest X-ray Negative for COVID-19\n n = 1022",
    x = rightx, y = 0.6, box_gp = gp, width = width))

connectGrob(numberwithxr, XRPos, "N")
connectGrob(numberwithxr, XRNeg, "N")

(RTPCRXRPos <- boxGrob("Chest X-Ray Positive with RT-PCR swab\n n = 625",
    x = leftx, y = 0.4, box_gp = gp, width = width))
(RTPCRXRNeg <- boxGrob("Chest X-Ray Negative with RT-PCR swab \n n = 573",
    x = rightx, y = 0.4, box_gp = gp, width = width))

connectGrob(XRPos, RTPCRXRPos, "N")
connectGrob(XRNeg, RTPCRXRNeg, "N")

(NoRTPCRXRPos <- boxGrob("No RT-PCR Swab\n n = 125",
    x = 0.4, y = 0.5, box_gp = gp, width = unit(1.5,
        "inch")))
(NoRTPCRXRNeg <- boxGrob("No RT-PCR Swab\n n = 449",
    x = 0.9, y = 0.5, box_gp = gp, width = unit(1.5,
        "inch")))

connectGrob(XRPos, NoRTPCRXRPos, "-")
connectGrob(XRNeg, NoRTPCRXRNeg, "-")

(MatchedXRPos <- boxGrob("Chest X-Ray Positive \nafter Propensity Score
         Matching\n n = 430",
    x = leftx, y = 0.225, box_gp = gp, width = width))
(MatchedXRNeg <- boxGrob("Chest X-Ray Negative \nafter Propensity Score
         Matching \n n = 430",
    x = 0.65, y = 0.25, box_gp = gp, width = unit(4.2,
        "inch")))

connectGrob(RTPCRXRPos, MatchedXRPos, "N")
connectGrob(RTPCRXRNeg, MatchedXRNeg, "N")
```

```
(UnmatchedXRPos <- boxGrob("Unmatched\n n = 195",
    x = 0.4, y = 0.325, box_gp = gp, width = unit(1.5,
        "inch")))
(UnmatchedXRNeg <- boxGrob("Unmatched\n n = 143",
    x = 0.9, y = 0.325, box_gp = gp, width = unit(1.5,
        "inch")))

connectGrob(RTPCRXRPos, UnmatchedXRPos, "-")
connectGrob(RTPCRXRNeg, UnmatchedXRNeg, "L")

(DiagXRPositive <- boxGrob("COVID-19 Positive n=305\n COVID-19 Negative n=125",
    x = leftx, y = 0.1, box_gp = gp, width = width))
(DiagXRNegative <- boxGrob("COVID-19 Positive n=243 \n COVID-19 Negative
         n=187",
    x = rightx, y = 0.1, box_gp = gp, width = width))

connectGrob(MatchedXRPos, DiagXRPositive,
    "N")
connectGrob(MatchedXRNeg, DiagXRNegative,
    "vertical")


(XRInd <- boxGrob("Chest X-Ray Indeterminate \n n = 197",
    x = 0.88, y = 0.25, box_gp = gp, width = unit(2.5,
        "inch")))

connectGrob(MatchedXRNeg, XRInd, "horizontal")

(DiagXRInd <- boxGrob("COVID-19 Positive n=136\n COVID-19 Negative n=63",
    x = 0.88, y = 0.17, box_gp = gp, width = unit(2,
        "inch")))
connectGrob(XRInd, DiagXRInd, "vertical")
```



```
##### CT Flow Chart####

grid.newpage()
(totalattendance <- boxGrob("Number of Patients Attending Emergency Department
         (ED) in Study Period\n n = 1862",
    x = midx, y = 0.9, box_gp = gp, width = 0.7))

(numberwithCT <- boxGrob("Total Number with Chest Computed Tompgraphy (CT)\n n
         = 319",
    x = midx, y = 0.75, box_gp = gp, width = width))
connectGrob(totalattendance, numberwithCT,
    "vertical")

(numberwithoutCT <- boxGrob("No Chest CT\n n = 1543",
    x = rightx, y = 0.825, box_gp = gp, width = unit(2,
        "inch"), height = 0.05))

connectGrob(totalattendance, numberwithoutCT,
    "-")


(CTPos <- boxGrob("CT Positive for COVID-19 \n n = 232",
    x = leftx, y = 0.6, box_gp = gp, width = width))
(CTNeg <- boxGrob("CT Negative for COVID-19\n n = 87",
    x = rightx, y = 0.6, box_gp = gp, width = width))

connectGrob(numberwithCT, CTPos, "N")
connectGrob(numberwithCT, CTNeg, "N")

(RTPCRCTPos <- boxGrob("CT Positive with RT-PCR swab\n n = 217",
    x = leftx, y = 0.4, box_gp = gp, width = width))
```

```
(RTPCRCTNeg <- boxGrob("CT Negative with RT-PCR swab \n n = 85",
    x = rightx, y = 0.4, box_gp = gp, width = width))

connectGrob(CTPos, RTPCRCTPos, "N")
connectGrob(CTNeg, RTPCRCTNeg, "N")

(NoRTPCRCTPos <- boxGrob("No RT-PCR Swab\n n = 15",
    x = 0.4, y = 0.5, box_gp = gp, width = unit(1.5,
        "inch")))
(NoRTPCRCTNeg <- boxGrob("No RT-PCR Swab\n n = 2",
    x = 0.9, y = 0.5, box_gp = gp, width = unit(1.5,
        "inch")))

connectGrob(CTPos, NoRTPCRCTPos, "-")
connectGrob(CTNeg, NoRTPCRCTNeg, "-")

(DiagCTPositive <- boxGrob("COVID-19 Positive n=162\n COVID-19 Negative n=55",
    x = leftx, y = 0.1, box_gp = gp, width = width))
(DiagCTNegative <- boxGrob("COVID-19 Positive n=29\n COVID-19 Negative n=56",
    x = rightx, y = 0.1, box_gp = gp, width = width))

connectGrob(RTPCRCTPos, DiagCTPositive, "N")
connectGrob(RTPCRCTNeg, DiagCTNegative, "N")


(CTInd <- boxGrob("CT Reported Indeterminate \n n = 31",
    x = 0.9, y = 0.275, box_gp = gp, width = unit(3,
        "inch")))

connectGrob(RTPCRCTNeg, CTInd, "N")

(DiagCTInd <- boxGrob("COVID-19 Positive n=16\n COVID-19 Negative n=15",
    x = 0.9, y = 0.17, box_gp = gp, width = unit(2,
        "inch")))
connectGrob(CTInd, DiagCTInd, "vertical")
```



```r
### Labels####
grid.newpage()
(indextest <- boxGrob("Index Tests", x = midx,
    y = 0.9, box_gp = gpar(fill = "light blue"),
    width = 0.7))

(reftest <- boxGrob("Index Tests and Reference Standards",
    x = midx, y = 0.4, box_gp = gpar(fill = "light blue"),
    width = 0.7))

(finaldiag <- boxGrob("Final Diagnoses",
    x = midx, y = 0.1, box_gp = gpar(fill = "light blue"),
    width = 0.7))
```

52 9 Pooled Regression after Multi…