# Solution to Series 7

**1.** **a)** $F_Y$ is shifted to the left compared to $F_X$.

**b)** We show it for $H_A$ and it follows for $H_0$ if we set $a = 0$. Under $H_A$: $F_X(x) = F_Y(x-a)$. The same holds for the densities, i.e. $f_X(x) = f_Y(x-a)$.
We need to show that the distribution of $D$ is symmetric around $a$, i.e. $P(D-a \leqslant d) = P(-(D-a) \leqslant d)$.
We have:

$$\begin{aligned} P(D-a \leqslant d) =& P(D \leqslant d+a) \\ =& P(X-Y \leqslant d+a) \\ =& \int P(X-Y \leqslant d+a | Y=y) f_Y(y)\,\mathrm{d}y \\ =& \int P(X-y \leqslant d+a | Y=y) f_Y(y)\,\mathrm{d}y \\ =& \int P(X-y \leqslant d+a) f_Y(y)\,\mathrm{d}y \end{aligned}$$

where last equality follows because of independence. Hence,

$$\begin{aligned} P(D-a \leqslant d) &= \int P(X-y \leqslant d+a) f_Y(y)\,\mathrm{d}y \\ &= \int P(X \leqslant d+a+y) f_Y(y)\,\mathrm{d}y \\ &= \int F_X(d+a+y) f_Y(y)\,\mathrm{d}y. \end{aligned}$$

Similarly, by exchanging the roles of $X$ and $Y$, we have

$$\begin{aligned} P(-(D-a) \leqslant d) =& P(-D \leqslant d-a) \\ =& P(Y-X \leqslant d-a) \\ =& \int F_Y(d-a+x) f_X(x)\,\mathrm{d}x. \end{aligned}$$

It is left to show that

$$\int F_X(d+a+y) f_Y(y)\,\mathrm{d}y = \int F_Y(d-a+x) f_X(x)\,\mathrm{d}x.$$

We use that $F_X(x) = F_Y(x-a)$ and $f_X(x) = f_Y(x-a)$, hence

$$\begin{aligned} \int F_X(d+a+y) f_Y(y)\,\mathrm{d}y &= \int F_Y(d+a+y-a) f_X(y+a)\,\mathrm{d}y \\ &= \int F_Y(d+y) f_X(y+a)\,\mathrm{d}y \\ &= \int F_Y(d-a+x) f_X(x)\,\mathrm{d}x \end{aligned}$$

where the last equality follows because of a change of variable $x = y + a$ and $\mathrm{d}x = \mathrm{d}y$.

**c)** Under $H_0$, we can permute the $X_i$ and $Y_i$ values for some $i$. We have the following four permutations: keeping the original group data, exchanging only $(X_1, Y_1)$, exchanging only $(X_2, Y_2)$, and exchanging both $(X_1, Y_1)$ and $(X_2, Y_2)$.
Exchanging only $(X_1, Y_1)$, i.e.

| | $X$ | $Y$ | $D$ | rank$(\lvert D\rvert)$ | $\mathbb{1}_{D_i>0}$ |
|---|---|---|---|---|---|
| $i=1$ | 0 | 0.5 | -0.5 | 1 | 0 |
| $i=2$ | 3 | 2 | 1 | 2 | 1 |

which results in $V = 1 \cdot 0 + 2 \cdot 1 = 2$.
Exchanging only $(X_2, Y_2)$, i.e.

| | $X$ | $Y$ | $D$ | rank$(\lvert D\rvert)$ | $\mathbb{1}_{D_i>0}$ |
|---|---|---|---|---|---|
| $i=1$ | 0.5 | 0 | 0.5 | 1 | 1 |
| $i=2$ | 2 | 3 | -1 | 2 | 0 |

which results in $V = 1 \cdot 1 + 2 \cdot 0 = 1$.
Exchanging both $(X_1, Y_1)$ and $(X_2, Y_2)$, i.e.

| | $X$ | $Y$ | $D$ | rank$(\lvert D\rvert)$ | $\mathbb{1}_{D_i>0}$ |
|---|---|---|---|---|---|
| $i=1$ | 0 | 0.5 | -0.5 | 1 | 0 |
| $i=2$ | 2 | 3 | -1 | 2 | 0 |

which results in $V = 1 \cdot 0 + 2 \cdot 0 = 0$.
In this case, the permutation distribution is a discrete uniform distribution on $\{0, 1, 2, 3\}$. We perform a one-sided test and larger values of $V$ indicate evidence towards $H_A$. The p-value is $P_{H_0}(V \geqslant 3) = 1/4$

**d)** Wilcoxon signed rank test

```
> (dat <- rbind(c(0.5, 0), c(3, 2)))
```

```
     [,1] [,2]
[1,]  0.5    0
[2,]  3.0    2
```

```
> wilcox.test(dat[, 1] - dat[, 2], alternative = "greater")
```

```
        Wilcoxon signed rank test

data:  dat[, 1] - dat[, 2]
V = 3, p-value = 0.25
alternative hypothesis: true location is greater than 0
```

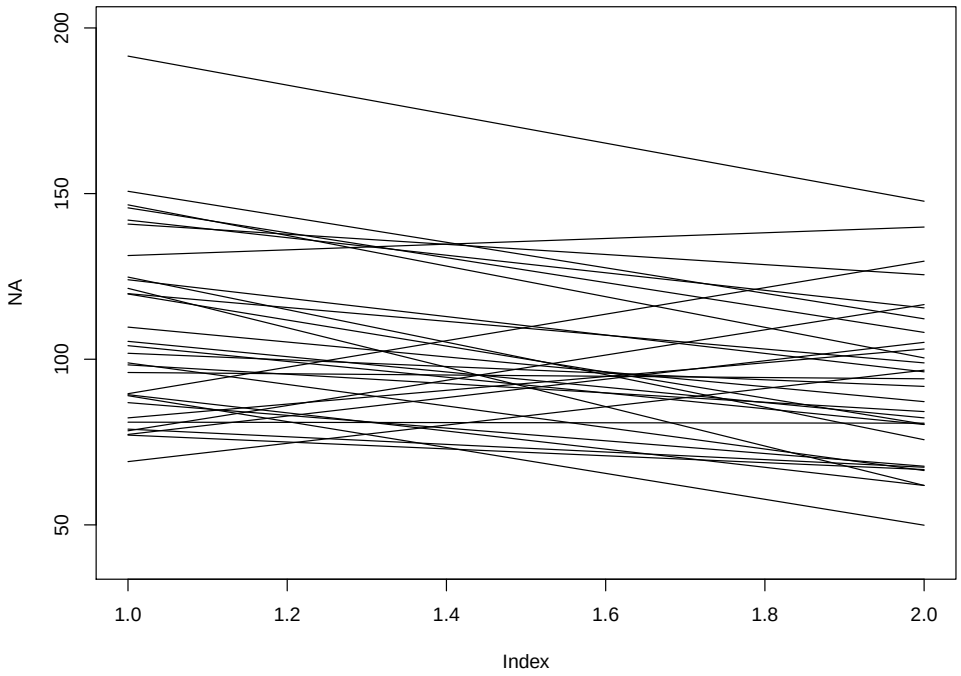
```
> # Alternative way to fit the Wilcoxon signed rank test.
> # wilcox.test(dat[, 1], dat[, 2], alternative = "greater",  paired = TRUE)
```

The values of the test statistic and the p-value are the same as the ones we computed by hand.

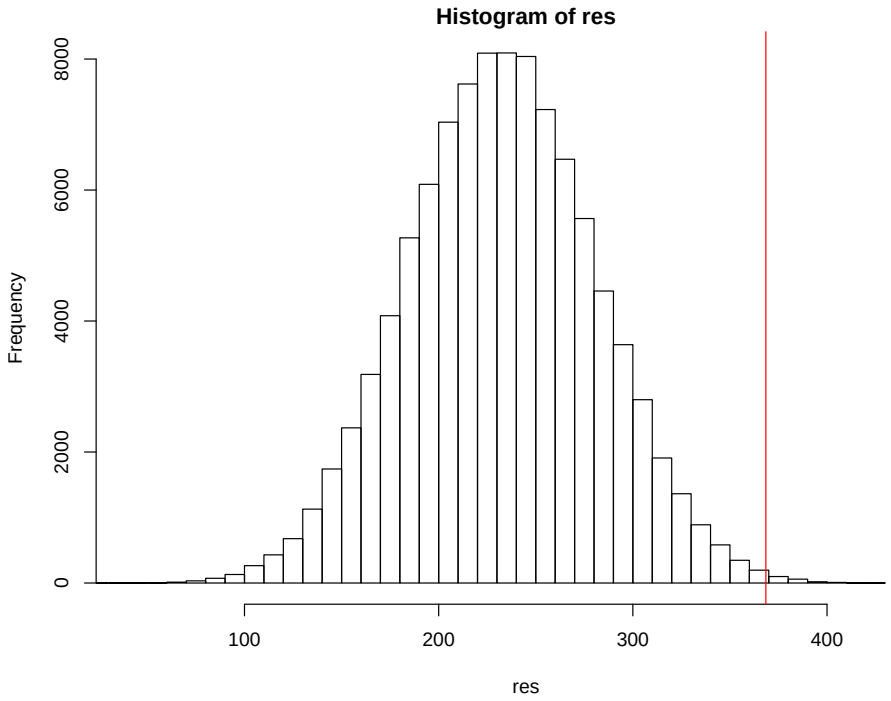
**2. a)**

```
> require("MASS")
> # Five varieties of barley were grown in six locations in each
> # of 1931 and 1932
> ?immer
> plot(x = NA, xlim = c(1, 2), ylim = c(40, 200))
> for (i in 1:nrow(immer)) {
  lines(x = c(1, 2), y = c(immer$Y1[i], immer$Y2[i]))
 }
> stem(immer$Y1 - immer$Y2)
```

```
  The decimal point is 1 digit(s) to the right of the |

  -4 | 0
  -3 | 8
  -2 | 881
  -1 |
  -0 | 9
   0 | 02
   1 | 002459
   2 | 1334778
   3 | 38899
   4 | 469
   5 |
   6 | 0
```




**b)**
```
> dd <- immer$Y1 - immer$Y2
> dd <- dd[dd != 0] # remove differences equal to zero
> dd.rank <- rank(abs(dd))
> dd.rank.sign <- dd.rank * sign(dd)
> (V <- sum(dd.rank.sign[dd.rank.sign > 0]))
[1] 368.5
> # Wilcoxon signed rank test
> Wilxocon.one.permutation <- function(y){
   n <- length(y)
   signs <- sample(c(-1, 1), n, replace = TRUE)
   d <- y * signs
   d.rank <- rank(abs(d))
   d.rank.sign <- d.rank * sign(d)
   ranks.pos <- sum(d.rank.sign[d.rank.sign > 0])
   return(ranks.pos)
 }
> set.seed(852)
> res <- replicate(100000, Wilxocon.one.permutation(dd))
> hist(res, breaks = 30, xlim = c(40, 450))
> abline(v = V, col = "red")
> (pval <- (sum(res >= V) + 1) / (length(res) + 1))
[1] 0.002119979
```




**c)** Wilcoxon signed rank test

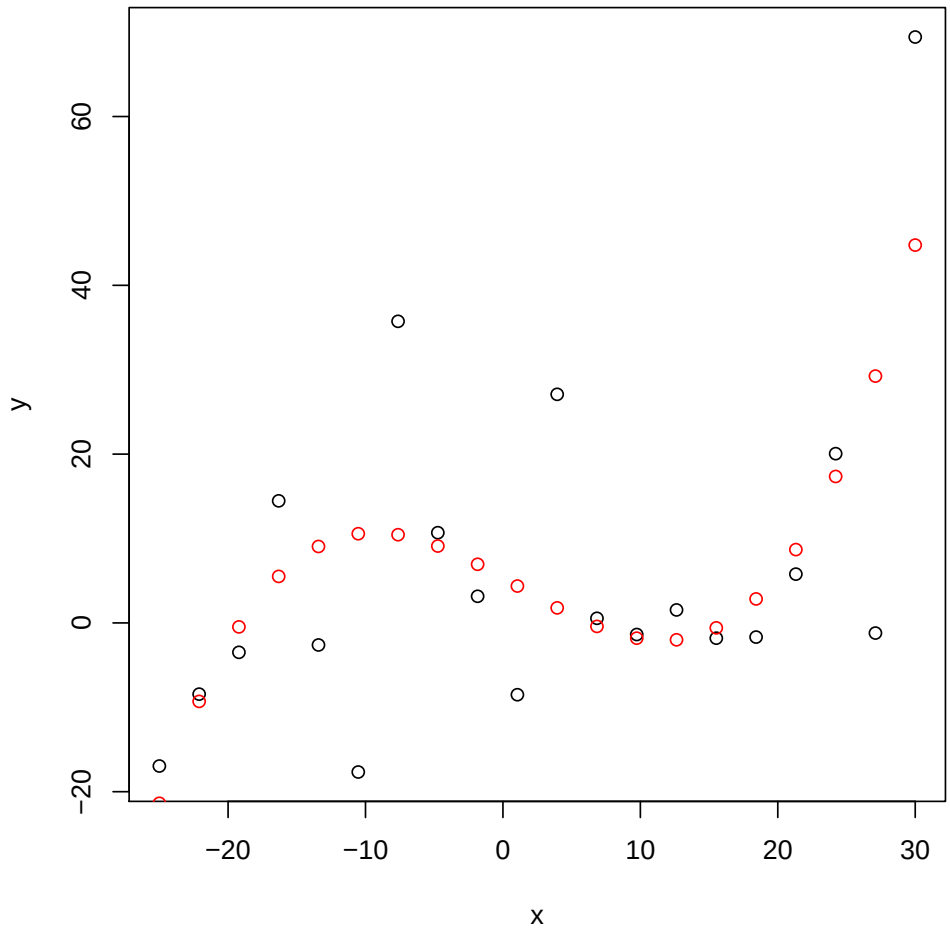
```
> # wilcox.test(immer$Y1, immer$Y2, alternative = "greater",  paired = TRUE)
> wilcox.test(immer$Y1 - immer$Y2, alternative = "greater")

        Wilcoxon signed rank test with continuity correction

data:  immer$Y1 - immer$Y2
V = 368.5, p-value = 0.002659
alternative hypothesis: true location is greater than 0
```

The values of the test statistic are the same and the p-values are very similar.

**3.** **a)**

```
> dat.org <- read.csv(file = "data_ex3.csv")
> # Fit regression model
> fit1  <- lm(y ~ x + I(x^2) + I(x^3), data = dat.org) # y ~ poly(x, 3)
> summary(fit1)

Call:
lm(formula = y ~ x + I(x^2) + I(x^3), data = dat.org)

Residuals:
     Min       1Q   Median       3Q      Max
-30.4469  -3.9743   0.6474   3.7564  25.3044

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.3310525  5.4735842   0.974   0.3446
x           -0.9048139  0.5282166  -1.713   0.1060
I(x^2)      -0.0094172  0.0162121  -0.581   0.5694
I(x^3)       0.0027799  0.0009924   2.801   0.0128 *
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 15.99 on 16 degrees of freedom
Multiple R-squared:  0.4613,        Adjusted R-squared:  0.3603
F-statistic: 4.568 on 3 and 16 DF,  p-value: 0.01706
```

```
> # plot fitted values on top
> plot(x = x, y = y)
> points(x = x, y = fitted(fit1), col = 2)
```




**b)** Calculate the p-value of the F test.

```
> (Ftest <- summary(fit1)$fstatistic)
```

```
    value     numdf     dendf
 4.567815  3.000000 16.000000
```

```
> 1 - pf(Ftest[1], df1 = Ftest[2], df2 = Ftest[3])
```

```
     value
0.01706132
```

```
> # # Alternatively
> # anova(lm(y ~ 1, data = dat.org), fit1)
```

**c)**

```
> # The following R code is used to calculate the type I error rate and the power.
> # The beta values are only used to simulate from the alternative, i.e. for the
> # calculation of the power.
> beta1 <- 0.5
> beta2 <- -0.003
> beta3 <- 0.0001
> n <- 20
> x <- seq(from = -25, to = 30, length.out = n)
> # Now do many simulations
> n.sim <- 250
> results.power <- numeric(n.sim)
> results.typeI <- numeric(n.sim)
> for(i in 1:n.sim){
   # simulate new response
```

```
    err <-  15 * (rgamma(n, shape = 2, rate = 1) - 2)
    y <- beta1 * x + beta2 * x^2 + beta3 * x^3 + err
    dat1 <- data.frame(y = y, x = x)

    # Result of global F-test
    # For every data set, we check whether the global null hypothesis
    # is being rejected or not.
    fit1  <- lm(y ~ x + I(x^2) + I(x^3), data = dat1)
    Ftest1 <- summary(fit1)$fstatistic
    pval1 <- 1 - pf(Ftest1[1], df1 = Ftest1[2], df2 = Ftest1[3])
    results.power[i] <- pval1 < 0.05

    # Result of global F-test under H_0
    # For every data set simulated under H_0, we check whether the F-test
    # is rejected.
    fit2  <- lm(y ~ x + I(x^2) + I(x^3),
                data = data.frame(y = err, x = x))
    Ftest2 <- summary(fit2)$fstatistic
    pval2 <- 1 - pf(Ftest2[1], df1 = Ftest2[2], df2 = Ftest2[3])
    results.typeI[i] <- pval2 < 0.05
  }
> # type I error
> mean(results.typeI)
[1] 0.052
```

**d)** See R code above.

```
> # power
> mean(results.power)
[1] 0.3
```

**e)**

```
> fit3 <- lm(y ~ x + I(x^2) + I(x^3), data = dat.org)
> observed.F <- summary(fit3)$fstatistic[1]
> # permutation test
> res.f <- rep(NA, 1000)
> dat.tmp <- dat.org
> for (i in 1:1000) {
    # permute only y
    dat.tmp$y <- dat.tmp$y[sample(1:nrow(dat.tmp), nrow(dat.tmp))]
    fit.tmp <- lm(y ~ x + I(x^2) + I(x^3), data = dat.tmp)
    res.f[i] <- summary(fit.tmp)$fstatistic[1]
  }
> # p-value
> (pval <- (sum(observed.F <= res.f) + 1) / (length(res.f) + 1))
[1] 0.02097902
```

**f)**

```
> # Note that the response has to be called y
> permutation_Ftest <- function(formula, data) {
    observed.F <- summary(lm(formula = formula, data = data))$fstatistic[1]

    # permutation test
    res.f <- rep(NA, 1000)
    for (i in 1:1000) {
      # permute only y
      data$y <- data$y[sample(1:nrow(data), nrow(data))]
      res.f[i] <- summary(lm(formula = formula, data = data))$fstatistic[1]
    }

    # p-value
    pval <- (sum(observed.F <= res.f) + 1) / (length(res.f) + 1)
```

```
   return(pval)
 }
> n.sim <- 250
> # Calculate the power using a permutation test
> results2.power <- numeric(n.sim)
> # Calculate the type I error rate using a permutation test
> results2.typeI <- numeric(n.sim)
> for(i in 1:n.sim){
   # simulate new response
   err <-  15 * (rgamma(n, shape = 2, rate = 1) - 2)
   y <- beta1 * x + beta2 * x^2 + beta3 * x^3 + err
   dat1 <- data.frame(y = y, x = x)

   # Result of global F-test
   # For every data set, we check whether the global null hypothesis
   # is being rejected or not.
   results2.power[i] <- permutation_Ftest(
     formula = as.formula(y ~ x + I(x^2) + I(x^3)),
     data = dat1) < 0.05

   # Result of global F-test under H_0
   # For every data set (simulated under H_0), we check whether the F-test
   # is rejected.
   results2.typeI[i] <- permutation_Ftest(
     formula = as.formula(y ~ x + I(x^2) + I(x^3)),
     data = data.frame(y = err, x = x)) < 0.05

   # print(i)
 }
>
> # power
> mean(results2.power)
[1] 0.324
> # type I error
> mean(results2.typeI)
[1] 0.032
```

The permutation test has slightly more power than the F-test under the given $H_A$.