1. United Nations life expectancy data

Life expectancy at birth is a measure of the average a living being is expected to live. It takes into account several demographic factors like gender, country, or year of birth.

Life expectancy at birth can vary along time or between countries because of many causes: the evolution of medicine, the degree of development of countries, or the effect of armed conflicts. Life expectancy varies between gender, as well. The data shows that women live longer that men. Why? Several potential factors, including biological reasons and the theory that women tend to be more health conscious.

Let's create some plots to explore the inequalities about life expectancy at birth around the world. We will use a dataset from the United Nations Statistics Division, which is available here (http://data.un.org/Data.aspx?d=GenderStat&f=inID:37&c=1,2,3,4,5,6&s=crEngName:asc,sgvEngName:asc,timeEngName:desc&v=1).

In [68]:
```
# This sets plot images to a nice size
options(repr.plot.width = 6, repr.plot.height = 6)

# Loading packages
library(dplyr)
library(tidyr)
library(ggplot2)

# Loading data
life_expectancy <- read.csv("datasets/UNdata.csv")

# Taking a look at the first few rows
head(life_expectancy)
```

A data.frame: 6 × 7

	Country.or.Area	Subgroup	Year	Source	Unit	Value	Value.Footnotes
	<chr>	<chr>	<chr>	<chr>	<chr>	<int>	<int>
1	Afghanistan	Female	2000-2005	UNPD_World Population Prospects_2006 (International estimate)	Years	42	NA
2	Afghanistan	Female	1995-2000	UNPD_World Population Prospects_2006 (International estimate)	Years	42	NA
3	Afghanistan	Female	1990-1995	UNPD_World Population Prospects_2006 (International estimate)	Years	42	NA
4	Afghanistan	Female	1985-1990	UNPD_World Population Prospects_2006 (International estimate)	Years	41	NA
5	Afghanistan	Male	2000-2005	UNPD_World Population Prospects_2006 (International estimate)	Years	42	NA
6	Afghanistan	Male	1995-2000	UNPD_World Population Prospects_2006 (International estimate)	Years	42	NA

In [69]:
```
# These packages need to be loaded in the first `@tests` cell.
library(testthat)
library(IRkernel.testthat)

# Then follows one or more tests of the students code.
# The @solution should pass the tests.
# The purpose of the tests is to try to catch common errors and to
# give the student a hint on how to resolve these errors.

run_tests({
  test_that("Test that life_expectancy exists", {
    expect_true(exists("life_expectancy"),
                info = "It seems that the data frame life_expectancy does not
exist.")
  })

  test_that("Test that life_expectancy is loaded correctly", {
    expect_true(nrow(life_expectancy)==1571,
                info = "The data frame life_expectancy is not correctly loade
d.")
  })

  test_that("Test that life_expectancy is loaded correctly", {
    expect_true(ncol(life_expectancy)==7,
                info = "The data frame life_expectancy is not correctly loade
d.")
  })
})
```

```
3/3 tests passed
```

2. Life expectancy of men vs. women by country

Let's manipulate the data to make our exploration easier. We will build the dataset for our first plot in which we will represent the average life expectancy of men and women across countries for the last period recorded in our data (2000-2005).

In [70]:
```
# Subsetting and reshaping the life expectancy data
subdata <- life_expectancy  %>%
    filter(Year == "2000-2005") %>%
    select(Country.or.Area, Subgroup, Value) %>%
    spread(Subgroup, Value)

# Taking a look at the first few rows
head(subdata)
```

A data.frame: 6 × 3

	Country.or.Area	Female	Male
	<chr>	<int>	<int>
1	Afghanistan	42	42
2	Albania	79	73
3	Algeria	72	70
4	Angola	43	39
5	Argentina	78	71
6	Armenia	75	68

In [71]:
```
# one or more tests of the students code.
# The @solution should pass the tests.
# The purpose of the tests is to try to catch common errors and to
# give the student a hint on how to resolve these errors.
run_tests({
  test_that("Test that subdata exists", {
    expect_true(exists("subdata"),
                info = "It seems that dataset subdata does not exist.")
  })

  test_that("Test that subdata is created correctly", {
    expect_true(nrow(subdata)==195,
                info = "It seems that subdata is not correctly created.")
  })

  test_that("Test that subdata is created correctly", {
    expect_true(ncol(subdata)==3,
                info = "It seems that subdata is not correctly created.")
  })

  test_that("Test that subdata is contains correct columns", {
    expect_true(sum(is.element(c("Country.or.Area", "Female", "Male"), names(s
ubdata)))==3,
                info = "It seems that subdata does not contain the correct co
lumns.")
  })
})
```

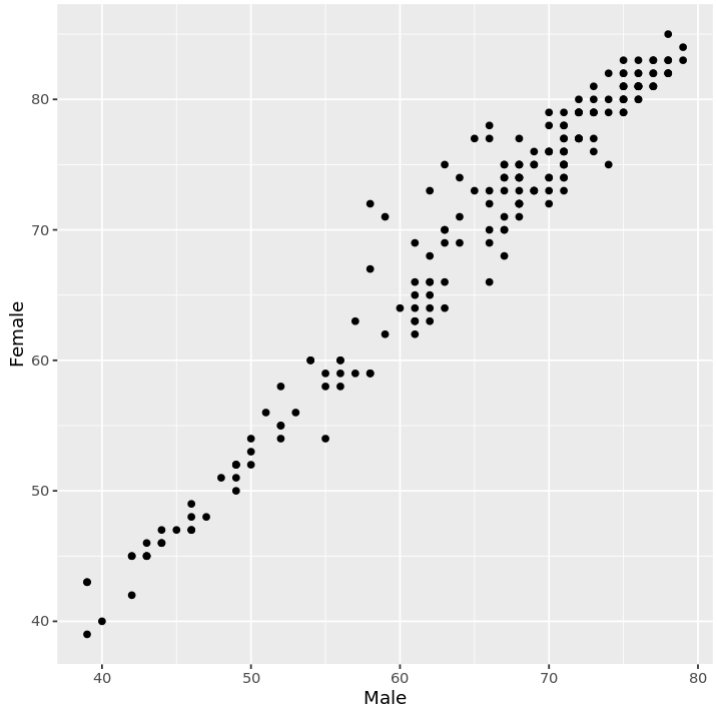
```
4/4 tests passed
```

3. Visualize I

A scatter plot is a useful way to visualize the relationship between two variables. It is a simple plot in which points are arranged on two axes, each of which represents one of those variables.

Let's create a scatter plot using `ggplot2` to represent life expectancy of males (on the x-axis) against females (on the y-axis). We will create a straightforward plot in this task, without many details. We will take care of these kinds of things shortly.

In [72]:
```
# Plotting male and female life expectancy
ggplot(subdata, aes(x = Male, y = Female)) +
geom_point()
```



In [73]:
```
run_tests({
  test_that("Check that a geom_point plot was plotted.", {
    expect_true( "GeomPoint" %in% class( last_plot()$layers[[1]]$geom ) ,
                 info = "Add geom_point() to produce a scatter plot.")
  })

  test_that("Check variables are correctly mapped.", {
    expect_true(deparse(last_plot()$mapping$x)=="~Male" & deparse(last_plot()$
mapping$y) == "~Female",
                 info = "Check that the variables are mapped to the correct ax
es.")
  })

})
```

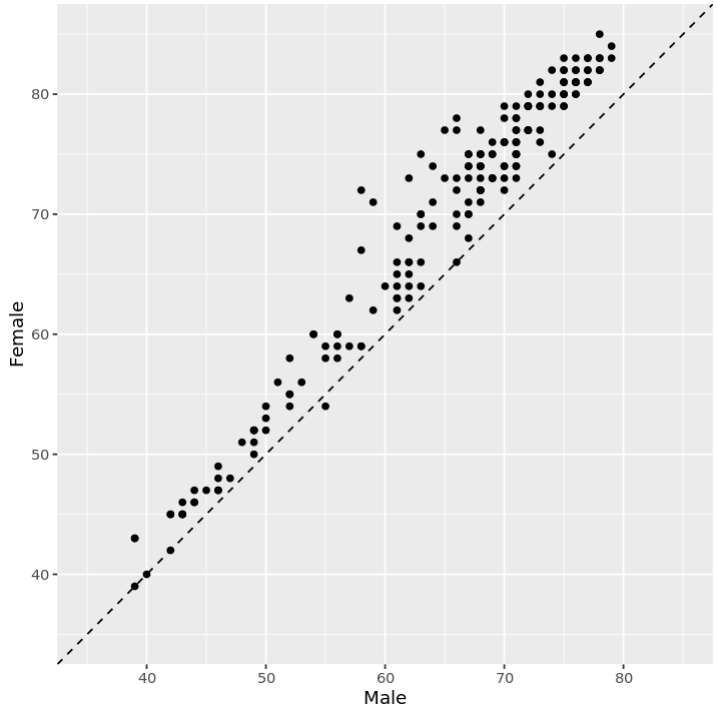
```
2/2 tests passed
```

4. Reference lines I

A good plot must be easy to understand. There are many tools in `ggplot2` to achieve this goal and we will explore some of them now. Starting from the previous plot, let's set the same limits for both axes as well as place a diagonal line for reference. After doing this, the difference between men and women across countries will be easier to interpret.

After completing this task, we will see how most of the points are arranged above the diagonal and how there is a significant dispersion among them. What does this all mean?

In [74]:
```
# Adding an abline and changing the scale of axes of the previous plots
ggplot(subdata, aes(x = Male, y = Female)) +
geom_point() +
geom_abline(linetype = "dashed", intercept = 0, slope = 1) +
xlim(35,85) +
ylim(35,85)
```



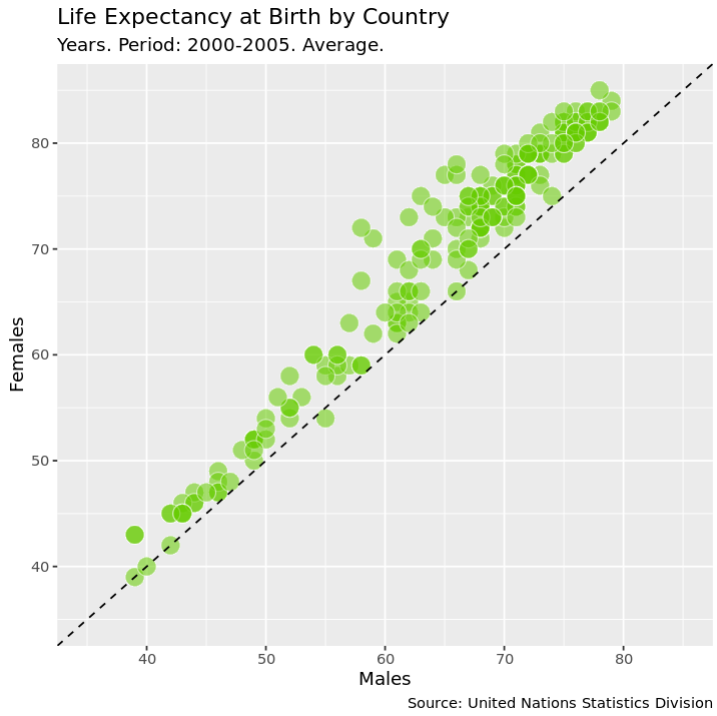
In [75]:
```
for (i in 1:length(ggplot_build(last_plot())$data))
{
  if ("slope" %in% colnames(ggplot_build(last_plot())$data[[i]])) i1=i
}


run_tests({
    test_that("Intercept of diagonal line is equal to 0.", {
    expect_equal(ggplot_build(last_plot())$data[[i1]]$intercept, 0,
        info = "Did you add the diagonal line correctly?")
    })
    test_that("Slope of diagonal line is equal to 1.", {
    expect_equal(ggplot_build(last_plot())$data[[i1]]$slope, 1,
        info = "Did you add the diagonal line correctly?")
    })
    test_that("Limits of x-axis.", {
        expect_equal(length(setdiff(c(39, 79), ggplot_build(last_plot())$layou
t$panel_scales_x[[1]]$range$range)), 0,
                info = "The limits of x-axis is not equal to [35, 85].")
})
    test_that("Limits of y-axis.", {
        expect_equal(length(setdiff(c(39, 85), ggplot_build(last_plot())$layou
t$panel_scales_y[[1]]$range$range)), 0,
                info = "The limits of y-axis is not equal to [35, 85].")
    })

})
```

```
4/4 tests passed
```

5. Plot titles and axis labels

A key point to make a plot understandable is placing clear labels on it. Let's add titles, axis labels, and a caption to refer to the source of data. Let's also change the appearance to make it clearer.

In [76]:
```
# Adding labels to previous plot
ggplot(subdata, aes(x = Male, y = Female)) +
  geom_point(color = "white", fill = "chartreuse3", shape = 21, alpha = .55, s
ize = 5) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed") +
  scale_x_continuous(limits = c(35,85)) +
  scale_y_continuous(limits = c(35,85)) +
  labs(title = "Life Expectancy at Birth by Country",
       subtitle = "Years. Period: 2000-2005. Average.",
       caption= "Source: United Nations Statistics Division",
       x= "Males",
       y= "Females")
```



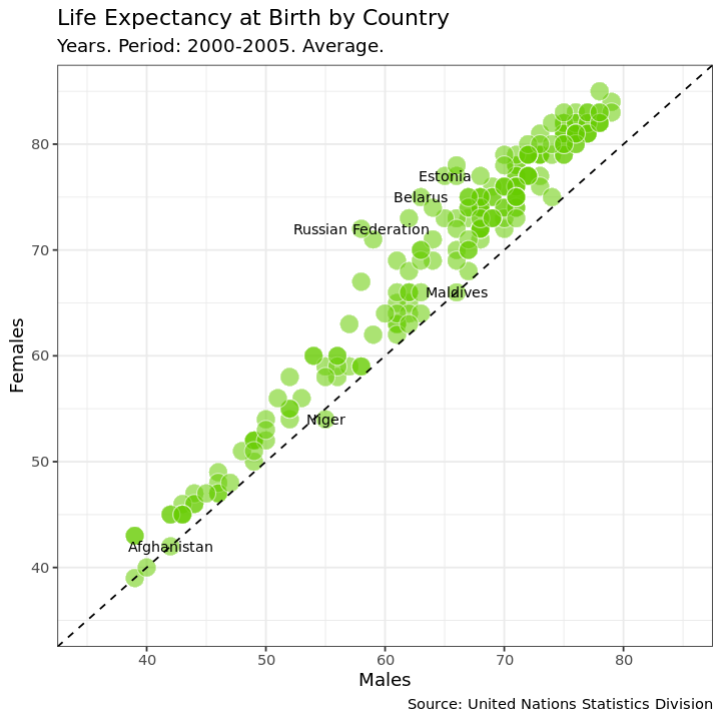
In [77]:
```
run_tests({
    test_that("Title is correct.", {
    expect_equal(toupper(gsub("[[:space:]]", "", last_plot()$labels$title)),
"LIFEEXPECTANCYATBIRTHBYCOUNTRY",
        info = "Did you add the title correctly?")
    })

        test_that("x-axis label is correct.", {
    expect_equal(toupper(gsub("[[:space:]]", "", last_plot()$labels$x)), "MALE
S",
        info = "Did you set the x-axis label correctly?")
    })


        test_that("y-axis label is correct.", {
    expect_equal(toupper(gsub("[[:space:]]", "", last_plot()$labels$y)), "FEMA
LES",
        info = "Did you set the y-axis label correctly?")
    })

          test_that("caption is correct.", {
    expect_equal(toupper(gsub("[[:space:]]", "", last_plot()$labels$caption)),
"SOURCE:UNITEDNATIONSSTATISTICSDIVISION",
        info = "Did you set the caption correctly?")
    })



})
```

```
4/4 tests passed
```

6. Highlighting remarkable countries I

Now, we will label some points of our plot with the name of its corresponding country. We want to draw attention to some special countries where the gap in life expectancy between men and women is significantly high. These will be the final touches on this first plot.

In [78]:
```
# Subseting data to obtain countries of interest
top_male <- subdata %>% arrange(Male-Female) %>% head(3)
top_female <- subdata %>% arrange(Female-Male) %>% head(3)

# Adding text to the previous plot to label countries of interest
ggplot(subdata, aes(x = Male, y = Female, label = Country.or.Area)) +
  geom_point(colour = "white", fill = "chartreuse3", shape = 21, alpha = .55,
size = 5)+
  geom_abline(intercept = 0, slope = 1, linetype = "dashed")+
  scale_x_continuous(limits = c(35,85)) +
  scale_y_continuous(limits = c(35,85)) +
  labs(title = "Life Expectancy at Birth by Country",
       subtitle = "Years. Period: 2000-2005. Average.",
       caption = "Source: United Nations Statistics Division",
       x = "Males",
       y = "Females") +
    geom_text(data = top_male, size = 3) +
    geom_text(data = top_female, size = 3) +
    theme_bw()
```



In [79]:
```
texts=c()
for (i in 1:length(last_plot()$layers)) texts=c(last_plot()$layers[[i]]$data$C
ountry.or.Area %>% as.character, texts)


run_tests({

      test_that("Test that countries defined by top_female and top_male are co
rrectly labeled.", {
    expect_true(length(setdiff(texts, c("Russian Federation", "Belarus", "Esto
nia", "Niger", "Afghanistan", "Maldives")))==0,
                  info = "It seems that countries defined by top_female and top
_male are not labeled correctly.")
  })

        test_that("Theme is theme_bw().", {
    expect_equal(last_plot()$theme$panel.background$fill, "white",
        info = "It seems that your plot does not have theme_bw().")
    })


})
```

```
2/2 tests passed
```

7. How has life expectancy by gender evolved?

Since our data contains historical information, let's see now how life expectancy has evolved in recent years. Our second plot will represent the difference between men and women across countries between two periods: 2000-2005 and 1985-1990.

Let's start building a dataset called `subdata2` for our second plot.

In [80]:
```
# Subsetting, mutating and reshaping the life expectancy data
subdata2 <- life_expectancy %>%
  filter(Year %in% c("1985-1990", "2000-2005")) %>%
  mutate(Sub_Year = paste(Subgroup, Year, sep = "_")) %>%
  mutate(Sub_Year = gsub("-", "_", Sub_Year)) %>%
  select(-Subgroup, -Year) %>%
  spread(Sub_Year, Value) %>%
  mutate(diff_Female = Female_2000_2005 - Female_1985_1990) %>%
  mutate(diff_Male = Male_2000_2005 - Male_1985_1990)

# Taking a look at the first few rows
head(subdata2)
```

A data.frame: 6 × 10

	Country.or.Area	Source	Unit	Value.Footnotes	Female_1985_1990	Female_2000_2005
	<chr>	<chr>	<chr>	<int>	<int>	<int>
1	Afghanistan	UNPD_World Population Prospects_2006 (International estimate)	Years	NA	41	42
2	Albania	UNPD_World Population Prospects_2006 (International estimate)	Years	NA	75	79
3	Algeria	UNPD_World Population Prospects_2006 (International estimate)	Years	NA	67	72
4	Angola	UNPD_World Population Prospects_2006 (International estimate)	Years	NA	42	43
5	Argentina	UNPD_World Population Prospects_2006 (International estimate)	Years	NA	75	78
6	Armenia	UNPD_World Population Prospects_2006 (International estimate)	Years	NA	71	75

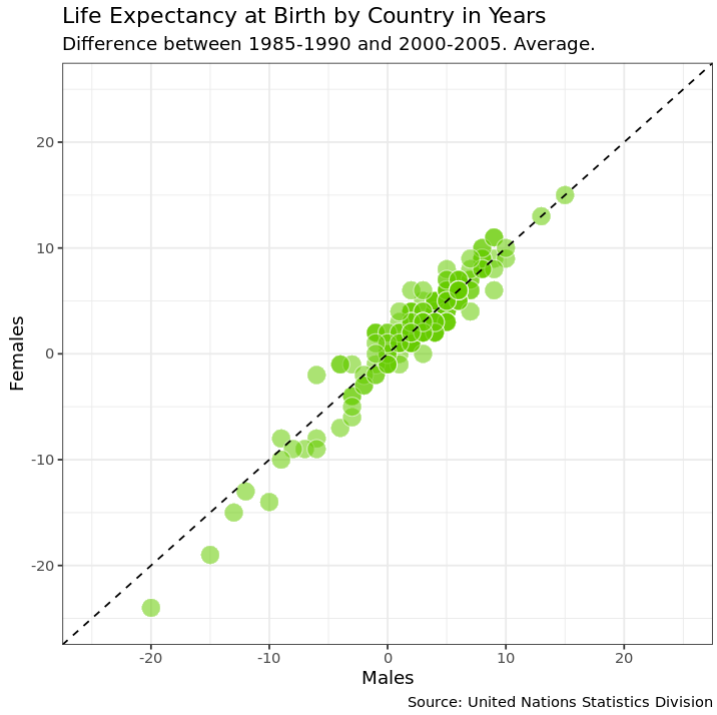
In [81]:
```
run_tests({
  test_that("Test that subdata2 is created correctly.", {
    expect_true(nrow(subdata2)==195,
                info = "It seems that dataset subdata2 is not correctly creat
ed.")
  })

  test_that("Test that subdata2 is created correctly.", {
    expect_true(ncol(subdata2)==10,
                info = "It seems that dataset subdata2 is not correctly creat
ed.")
  })

  test_that("Test that subdata2 is created correctly.", {
    expect_true(length(setdiff(c('diff_Female', 'diff_Male'), names(subdata2
)))==0,
                info = "It seems that subdata2 does not contain columns diff_
Female or diff_Male.")
  })

    test_that("Test that subdata2 is created correctly.", {
    expect_true(sum(subdata2$diff_Female)==492,
                info = "It seems that the diff_Female column is not correctly
created.")
  })

  test_that("Test that subdata2 is created correctly.", {
    expect_true(sum(subdata2$diff_Male)==503,
                info = "It seems that the diff_Male column is not correctly c
reated.")
  })
})
```

```
5/5 tests passed
```

8. Visualize II

Now let's create our second plot in which we will represent average life expectancy differences between "1985-1990" and "2000-2005" for men and women.

In [82]:
```
# Doing a nice first version of the plot with abline, scaling axis and adding
 labels
ggplot(subdata2, aes(x = diff_Male, y= diff_Female, label = Country.or.Area))
+
  geom_point(colour = "white", fill = "chartreuse3", shape = 21, alpha = .55,
size = 5) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed")+
  scale_x_continuous(limits = c(-25, 25)) +
  scale_y_continuous(limits = c(-25, 25)) +
  labs(title="Life Expectancy at Birth by Country in Years",
       subtitle="Difference between 1985-1990 and 2000-2005. Average.",
       caption="Source: United Nations Statistics Division",
       x="Males",
       y="Females") +
theme_bw()
```



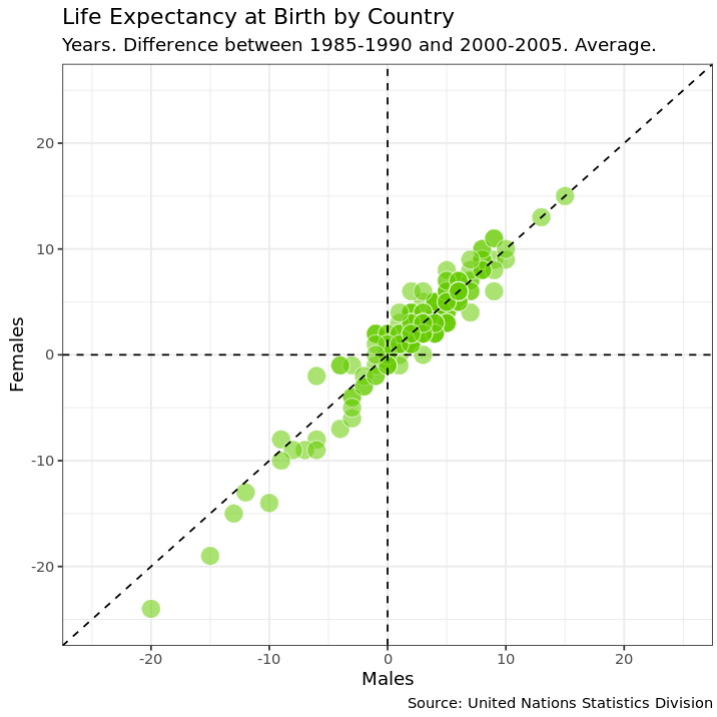
In [83]:
```
run_tests({

#    test_that("Check that a geom_point plot was plotted.", {
#     expect_true( "GeomPoint" %in% class( last_plot()$layers[[1]]$geom ) ,
#                  info = "Add geom_point() to produce a scatter plot.")
#   })

  test_that("Check variables are correctly mapped.", {
    expect_true( deparse(last_plot()$mapping$x)=="~diff_Male" & deparse(last_p
lot()$mapping$y)=="~diff_Female",
                 info = "Check that the variables are mapped to the correct ax
es.")
  })


#     test_that("Intercept of diagonal line is equal to 0.", {
#     expect_equal(ggplot_build(last_plot())$data[[2]]$intercept, 0,
#         info = "Did you add the diagonal line correctly?")
#     })
#     test_that("Slope of diagonal line is equal to 1.", {
#     expect_equal(ggplot_build(last_plot())$data[[2]]$slope, 1,
#         info = "Did you add the diagonal line correctly?")
#     })
    test_that("Limits of x-axis", {
        expect_equal(length(setdiff(c(-20, 15), ggplot_build(last_plot())$layo
ut$panel_scales_x[[1]]$range$range)), 0,
                info = "Limits of x-axis is not equal to [-25, 25].")
    })
    test_that("Limits of y-axis", {
        expect_equal(length(setdiff(c(-24, 15), ggplot_build(last_plot())$layo
ut$panel_scales_y[[1]]$range$range)), 0,
                info = "Limits of y-axis is not equal to [-25, 25]")
    })

#     test_that("Intercept of diagonal line is equal to 0.", {
#     expect_equal(toupper(gsub("[[:space:]]", "", last_plot()$labels$title)),
"LIFEEXPECTANCYATBIRTHBYCOUNTRYINYEARS",
#         info = "Did you add the title correctly?")
#     })
#     test_that("Slope of diagonal line is equal to 1.", {
#     expect_equal(last_plot()$theme$panel.background$fill, "white",
#         info = "It seems that your plot does not have theme_bw().")
#     })
})
```

```
3/3 tests passed
```

9. Reference lines II

Adding reference lines can make plots easier to understand. We already added a diagonal line to visualize differences between men and women more clearly. Now we will add two more lines to help to identify in which countries people increased or decreased their life expectancy in the period analyzed.

In [84]:
```
# Adding an hline and vline to previous plots
ggplot(subdata2, aes(x=diff_Male, y=diff_Female, label=Country.or.Area))+
  geom_point(colour="white", fill="chartreuse3", shape=21, alpha=.55, size=5)+
  geom_abline(intercept = 0, slope = 1, linetype=2)+
  scale_x_continuous(limits=c(-25,25))+
  scale_y_continuous(limits=c(-25,25))+
  geom_hline(yintercept = 0, linetype = "dashed") +
  geom_vline(xintercept = 0, linetype = "dashed") +
  labs(title="Life Expectancy at Birth by Country",
       subtitle="Years. Difference between 1985-1990 and 2000-2005. Average.",
       caption="Source: United Nations Statistics Division",
       x="Males",
       y="Females")+
theme_bw()
```



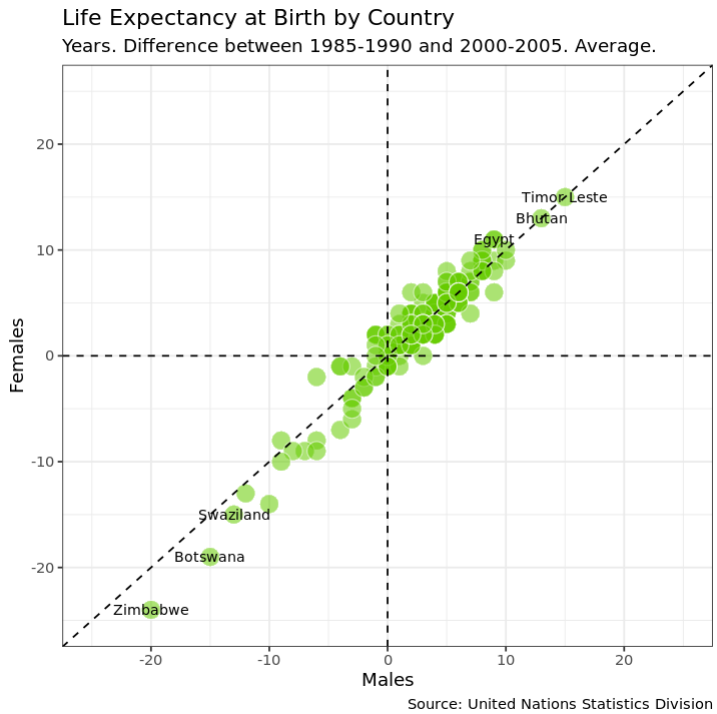
In [85]:
```
for (i in 1:length(ggplot_build(last_plot())$data))
{
  if ("slope"      %in% colnames(ggplot_build(last_plot())$data[[i]])) i1=i
  if ("yintercept" %in% colnames(ggplot_build(last_plot())$data[[i]])) i2=i
  if ("xintercept" %in% colnames(ggplot_build(last_plot())$data[[i]])) i3=i
}


run_tests({
#     test_that("Intercept of diagonal line is equal to 0.", {
#     expect_equal(ggplot_build(last_plot())$data[[i1]]$intercept, 0,
#         info = "Did you add the diagonal line correctly?")
#     })
#     test_that("Slope of diagonal line is equal to 1.", {
#     expect_equal(ggplot_build(last_plot())$data[[i1]]$slope, 1,
#         info = "Did you add the diagonal line correctly?")
#     })

        test_that("Horizontal line is well defined.", {
    expect_equal(ggplot_build(last_plot())$data[[i2]]$yintercept, 0,
        info = "Did you add the horizontal line correctly?")
    })
    test_that("Vertical line is well defined.", {
    expect_equal(ggplot_build(last_plot())$data[[i3]]$xintercept, 0,
        info = "Did you add the vertical line correctly?")
    })
})
```

```
2/2 tests passed
```

10. Highlighting remarkable countries II

As we did in the first plot, let's label some points. Concretely, we will point those three where the aggregated average life expectancy for men and women increased most and those three where decreased most in the period.

In [86]:
```
# Subseting data to obtain countries of interest
top <- subdata2 %>% arrange(diff_Male+diff_Female) %>% head(3)
bottom <- subdata2 %>% arrange(-(diff_Male+diff_Female)) %>% head(3)

# Adding text to the previous plot to label countries of interest
ggplot(subdata2, aes(x = diff_Male, y = diff_Female, label = Country.or.Area),
guide = FALSE) +
  geom_point(colour = "white", fill = "chartreuse3", shape = 21, alpha = .55,
size = 5) +
  geom_abline(intercept = 0, slope = 1, linetype = 2) +
  scale_x_continuous(limits = c(-25, 25)) +
  scale_y_continuous(limits = c(-25, 25)) +
  geom_hline(yintercept = 0, linetype = "dashed")+
  geom_vline(xintercept = 0, linetype = "dashed")+
  labs(title="Life Expectancy at Birth by Country",
       subtitle="Years. Difference between 1985-1990 and 2000-2005. Average.",
       caption="Source: United Nations Statistics Division",
       x="Males",
       y="Females") +
  geom_text(data = top, size = 3) +
  geom_text(data = bottom, size = 3) +
  theme_bw()
```



In [87]:

```
texts=c()
for (i in 1:length(last_plot()$layers)) texts=c(last_plot()$layers[[i]]$data$C
ountry.or.Area %>% as.character, texts)

run_tests({
  test_that("Test that dataset bottom exists.", {
    expect_true(exists("bottom"),
                info = "It seems that bottom does not exist.")
  })

  test_that("Test that dataset bottom is correctly created.", {
    expect_true(nrow(bottom)==3,
                info = "It seems that bottom is not correctly created.")
  })

  test_that("Test that countries defined by top and bottom are correctly label
ed.", {
    expect_true(length(setdiff(texts, c("Timor Leste", "Bhutan", "Egypt", "Zim
babwe", "Botswana", "Swaziland")))==0,
                info = "It seems that countries defined by top and bottom are
not labeled correctly.")
  })


})
```

```
3/3 tests passed
```