```r
In [86]: # Load the required library
library(ggplot2)
library(dplyr)
library(stringr)
```

Data Cleaning

```r
In [87]: # Step 1: Load the dataset
df <- read.csv("Citywide_Payroll_Data__Fiscal_Year_.csv")
```

```r
In [88]: new_col <- c("Fiscal_Year", "Agency_Name", "Last_Name", "First_Name", "Mid_Init",
                     "Agency_Start_Date", "Work_Location_Borough", "Title_Description",
                     "Leave_Status_as_of_June_30", "Base_Salary", "Pay_Basis",
                     "Regular_Hours", "Regular_Gross_Paid", "OT_Hours",
                     "Total_OT_Paid", "Total_Other_Pay")

names(df) <- new_col
```

```r
In [89]: str(df)
```

```
'data.frame':   2194488 obs. of  16 variables:
 $ Fiscal_Year               : int  2016 2016 2016 2016 2016 2016 2016 2016 2016 2016 ...
 $ Agency_Name               : Factor w/ 165 levels "ADMIN FOR CHILDREN'S SVCS     ",..: 85 85 85 85 85 85 85 85 85 85 ...
 $ Last_Name                 : Factor w/ 135712 levels ""," CARMONA GARCIA",..: 33 436 551 569 597 621 621 790 888 894 ...
 $ First_Name                : Factor w/ 73183 levels ""," ARIA"," BLANCA",..: 51348 40231 3196 29679 29922 28172 30192 28172 41416 53119 ...
 $ Mid_Init                  : Factor w/ 44 levels "","-","\"","(",..: 22 1 29 27 1 1 21 1 27 22 ...
 $ Agency_Start_Date         : Factor w/ 13871 levels "01/01/1901","01/01/1969",..: 7544 6305 10948 12862 5520 9344 9341 6105 9250 5843 ...
 $ Work_Location_Borough     : Factor w/ 28 levels "","ALBANY       ",..: 13 13 13 13 13 13 13 13 13 13 ...
 $ Title_Description         : Factor w/ 1964 levels "","                                                            ",..: 654 121 654 680 616 280 280 616 654 1445
...
 $ Leave_Status_as_of_June_30: Factor w/ 5 levels "ACTIVE","CEASED",..: 1 1 3 1 2 1 1 1 2 1 ...
 $ Base_Salary               : Factor w/ 76321 levels "$0.50","$1.00",..: 36394 8507 29496 7117 2 56899 69249 2 30456 52449 ...
 $ Pay_Basis                 : Factor w/ 8 levels " per Annum"," per Day",..: 1 1 1 1 3 1 1 3 1 1 ...
 $ Regular_Hours             : num  1830 1831 1182 1831 0 ...
 $ Regular_Gross_Paid        : Factor w/ 1076134 levels "$-0.01","$-0.03",..: 618434 75899 354110 63504 462023 858312 998936 648869 1049100 792215 ...
 $ OT_Hours                  : num  2.25 0 1 0 0 0 0 0 0 0 ...
 $ Total_OT_Paid             : Factor w/ 492187 levels "$-0.01","$-0.02",..: 388710 553 166655 553 553 553 553 553 553 553 ...
 $ Total_Other_Pay           : Factor w/ 479574 levels "$-0.01","$-0.02",..: 11276 11276 280849 11276 11276 11276 428713 11276 11276 303390 ...
```

```r
In [90]: #Convert data types
df$Fiscal_Year <- as.numeric(df$Fiscal_Year)
df$Agency_Start_Date <- as.Date(df$Agency_Start_Date, format = "%m/%d/%Y")
df$Base_Salary <- as.numeric(gsub("[$,]", "", df$Base_Salary))
df$Regular_Gross_Paid <- as.numeric(gsub("[$,]", "", df$Regular_Gross_Paid))
df$Total_OT_Paid <- as.numeric(gsub("[$,]", "", df$Total_OT_Paid))
df$Total_Other_Pay <- as.numeric(gsub("[$,]", "", df$Total_Other_Pay))
```

```r
In [91]: #Check for and remove duplicate rows (if needed)
df <- distinct(df)

#Check the cleaned dataset
head(df)
```

Fiscal_Year	Agency_Name	Last_Name	First_Name	Mid_Init	Agency_Start_Date	Work_Location_Borough	Title_Description	Leave_Status_as_of_June_30	Base_Salary	Pay_Basis	Regular_Hours	Regular_G
2016	DISTRICT ATTORNEY-MANHATTAN	ABA'AHMID	RAHASHEEM	E	2003-07-14	MANHATTAN	COMMUNITY ASSOCIATE	ACTIVE	47678	per Annum	1830.00	
2016	DISTRICT ATTORNEY-MANHATTAN	ABENSUR	MARGARET		1995-06-12	MANHATTAN	ADMINISTRATIVE ACCOUNTANT	ACTIVE	119959	per Annum	1831.00	
2016	DISTRICT ATTORNEY-MANHATTAN	ABOUNAOUM	ANDREA	L	2011-10-11	MANHATTAN	COMMUNITY ASSOCIATE	ON LEAVE	39966	per Annum	1181.68	
2016	DISTRICT ATTORNEY-MANHATTAN	ABRAHAM	JONATHAN	J	2014-12-01	MANHATTAN	COMPUTER SYSTEMS MANAGER	ACTIVE	116000	per Annum	1831.00	
2016	DISTRICT ATTORNEY-MANHATTAN	ABRAMS	JOSEPH		2015-05-21	MANHATTAN	COLLEGE AIDE	CEASED	1	per Hour	0.00	
2016	DISTRICT ATTORNEY-MANHATTAN	ABREU	JENNIFER		2012-09-04	MANHATTAN	ASSISTANT DISTRICT ATTORNEY	ACTIVE	71500	per Annum	1831.00	

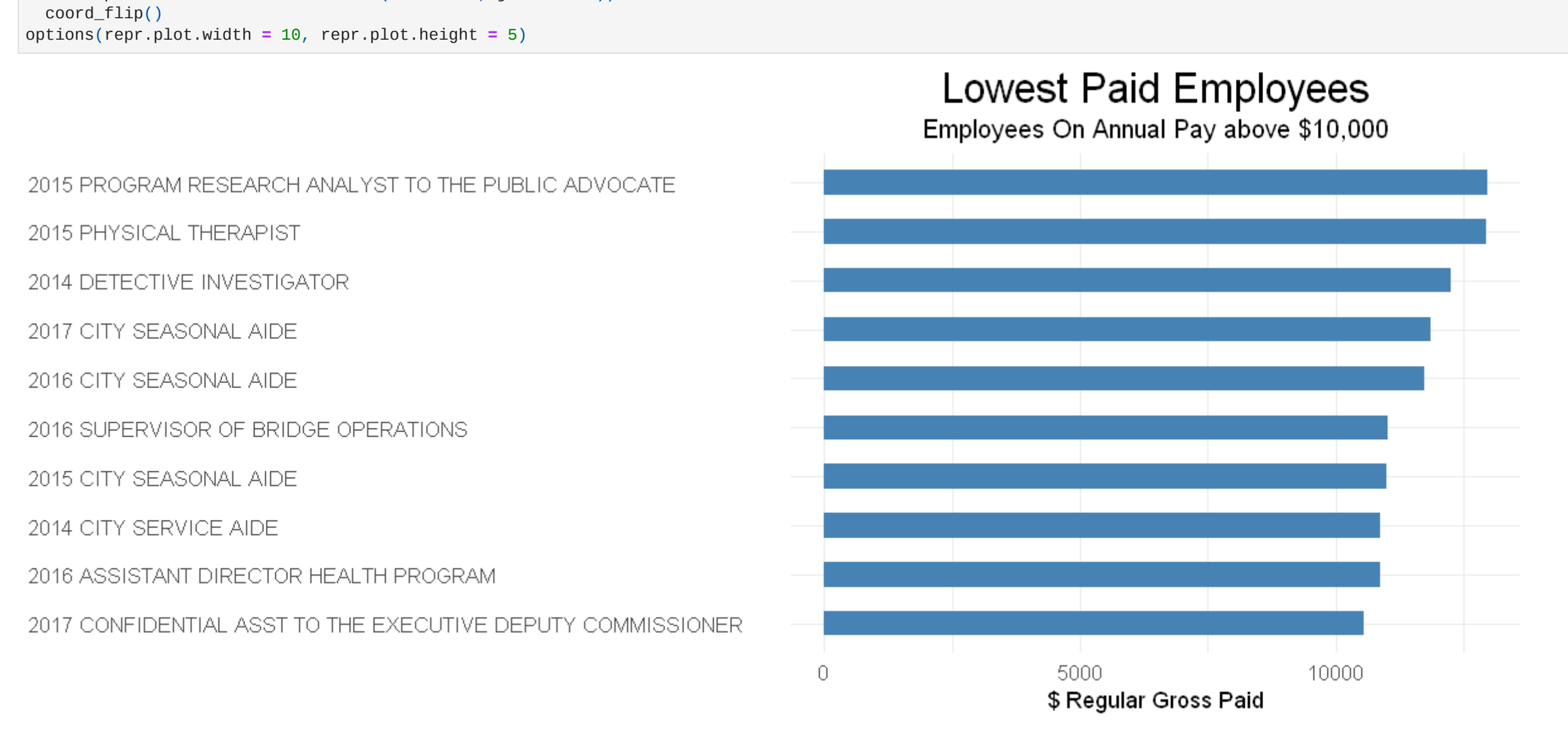
Lowest Paid Employees

According to the displayed chart, it seems that city aides have the lowest average paid compared to all other employees. City aides are present four times within the list of the ten lowest paid employees across the years 2014, 2015, 2016, and 2017. While there was a slight increase in the average paid for city aides in 2016 and 2017, it remains as one of the least among all roles. Except for the role of city aide, all other positions appeared only once. This indicates that their average pay increased after the initial year when they were among the lowest paid. Following this increase, their average pay improved to the extent that they no longer remained on the list.

```r
In [7]: # Subset of Data, due to pay having negative values and zeros.
x <- df %>%
  filter(`Regular_Gross_Paid` > 10000 & `Base_Salary` > 10000)

# Calculate the mean Regular_Gross_Paid for each unique Title_Description and Fiscal_Year
mean_data <- x %>%
  group_by(`Title_Description`, `Fiscal_Year`) %>%
  summarize(mean_Regular_Gross_Paid = mean(`Regular_Gross_Paid`))

# Sort the data by mean_Regular_Gross_Paid in ascending order
sorted_data <- mean_data %>%
  arrange(mean_Regular_Gross_Paid)

# Display the top 10 lowest mean_Regular_Gross_Paid
top_10_lowest <- head(sorted_data, 10)
top_10_lowest$Title_Description <- sub("^\\?", "", top_10_lowest$Title_Description)

# Print the result with the specified format
print(top_10_lowest)
```

```
# A tibble: 10 x 3
# Groups:   Title_Description [8]
   Title_Description                          Fiscal_Year mean_Regular_Gross_~
   <chr>                                            <dbl>                <dbl>
 1 "CONFIDENTIAL ASST TO THE EXECUTIVE DEPUTY ~        2017               10548.
 2 "ASSISTANT DIRECTOR HEALTH PROGRAM         ~        2016               10863.
 3 "CITY SERVICE AIDE                         ~        2014               10867.
 4 "CITY SEASONAL AIDE                        ~        2015               11008.
 5 "SUPERVISOR OF BRIDGE OPERATIONS           ~        2016               11012.
 6 "CITY SEASONAL AIDE                        ~        2016               11739.
 7 "CITY SEASONAL AIDE                        ~        2017               11851.
 8 "DETECTIVE INVESTIGATOR                    ~        2014               12265.
 9 "PHYSICAL THERAPIST                        ~        2015               12954.
10 "PROGRAM RESEARCH ANALYST TO THE PUBLIC ADV~        2015               12970.
```

```r
In [92]: #creating the plot
ggplot(top_10_lowest, aes(x = reorder(paste(Fiscal_Year, Title_Description), mean_Regular_Gross_Paid),
                          y = mean_Regular_Gross_Paid)) +
  geom_bar(stat = "identity", fill = "steelblue", width = 0.5) +
  labs(x = NULL, y = "$ Regular Gross Paid",title = "Lowest Paid Employees",
       subtitle = "Employees On Annual Pay above $10,000") +
  theme_minimal() +
  theme(axis.text.x = element_text(size = 10),
        axis.text.y = element_text(size = 10, hjust = 0),  # Set hjust for alignment
        axis.ticks.y = NULL,
        plot.margin = margin(5, 5, 50, 5, "pt"),
        plot.title = element_text(size = 20,hjust = 0.5),
        plot.subtitle = element_text(size = 12,hjust = 0.5)) +
  coord_flip()
options(repr.plot.width = 10, repr.plot.height = 5)
```



Highest Paid Employees

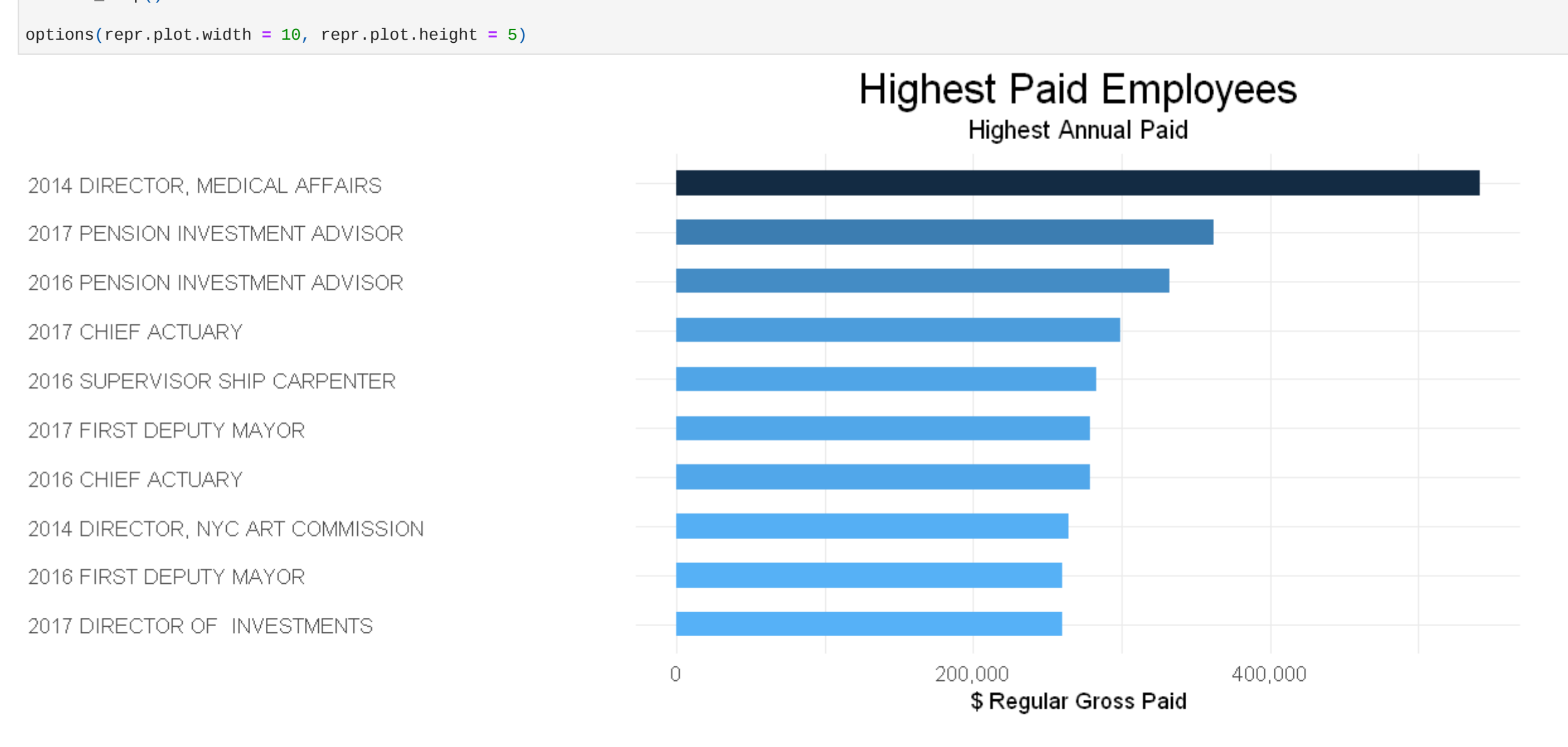
The highest paid role observed across the years 2014 to 2017 was the Director of Medical Affairs, as seen in 2014. However, this role only made it to the top ten highest-paid employees list once during this period. This suggests that after 2014, the compensation for this position decreased, causing it to no longer feature among the top ten highest-paid employees.

Subsequently, we find the role of Pension Investment Advisor, which appeared twice in both 2016 and 2017. Similarly, Chief Actuary and First Deputy Mayor each made two appearances on the list in 2016 and 2017. It can be inferred that the most consistent and highest-paid roles in 2016 and 2017 were the Pension Investment Advisor, followed by Chief Actuary, and then First Deputy Mayor.

```r
In [9]: mean_data <- df %>%
  group_by(`Title_Description`, `Fiscal_Year`) %>%
  summarize(mean_Regular_Gross_Paid = mean(`Regular_Gross_Paid`))

# Sort the data by mean_Regular_Gross_Paid in ascending order
sorted_data <- mean_data %>%
  arrange(desc(mean_Regular_Gross_Paid))

# Display the top 10 highest mean_Regular_Gross_Paid
top_10_highest <- head(sorted_data, 10)
top_10_highest$Title_Description <- sub("^\\?", "", top_10_highest$Title_Description)

# Print the result with the specified format
print(top_10_highest)
```

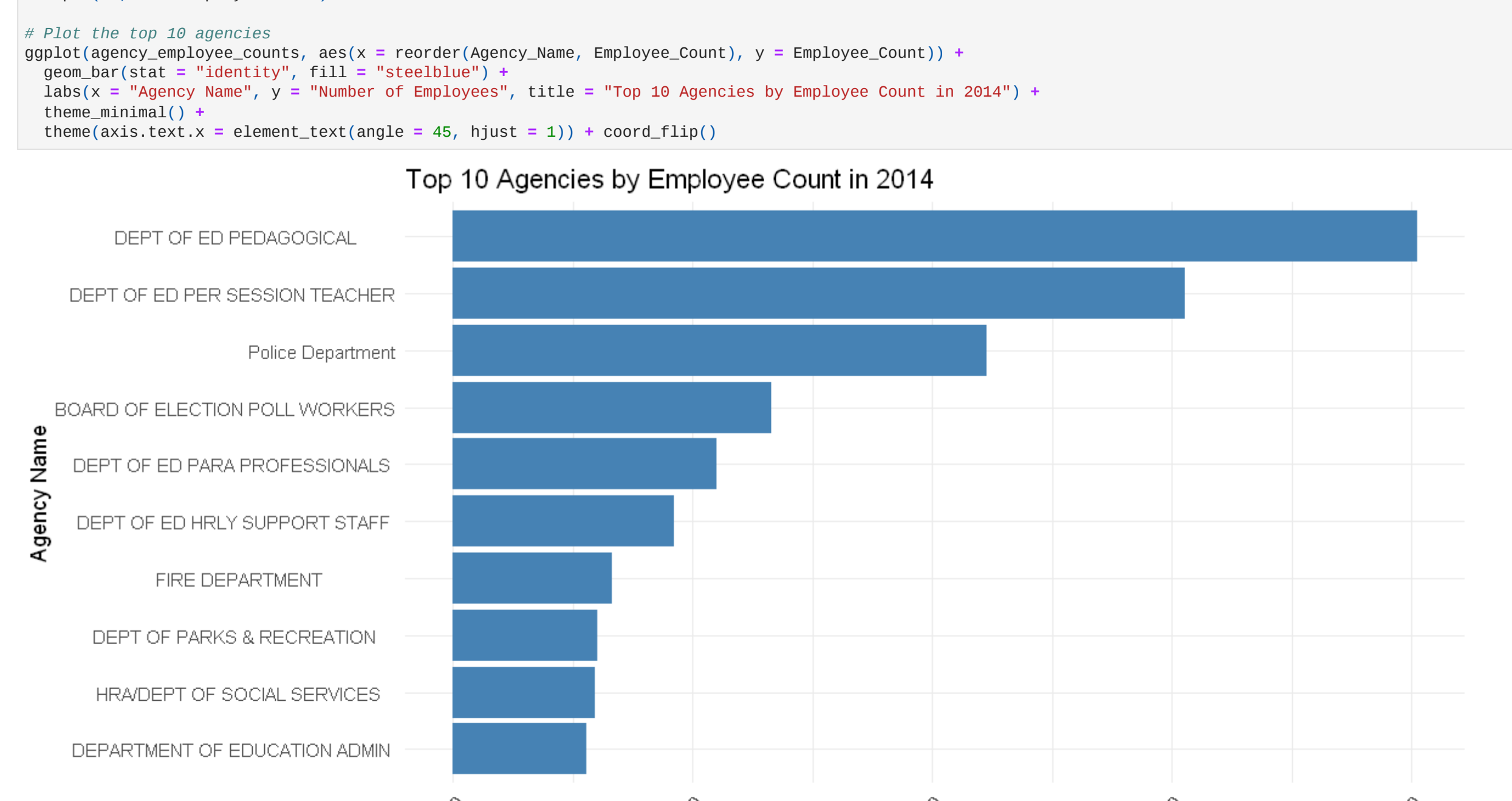
```
# A tibble: 10 x 3
# Groups:   Title_Description [7]
   Title_Description                          Fiscal_Year mean_Regular_Gross_~
   <chr>                                            <dbl>                <dbl>
 1 "DIRECTOR, MEDICAL AFFAIRS                 ~        2014              541545.
 2 "PENSION INVESTMENT ADVISOR                ~        2017              362416.
 3 "PENSION INVESTMENT ADVISOR                ~        2016              332081.
 4 "CHIEF ACTUARY                             ~        2017              298755.
 5 "SUPERVISOR SHIP CARPENTER                 ~        2016              282772.
 6 "FIRST DEPUTY MAYOR                        ~        2017              279003.
 7 "CHIEF ACTUARY                             ~        2016              278886.
 8 "DIRECTOR, NYC ART COMMISSION              ~        2014              264307.
 9 "FIRST DEPUTY MAYOR                        ~        2016              260447.
10 "DIRECTOR OF  INVESTMENTS                  ~        2017              259650.
```

```r
In [10]: #creating the plot
ggplot(top_10_highest, aes(x = reorder(paste(Fiscal_Year, Title_Description), mean_Regular_Gross_Paid),
                           y = mean_Regular_Gross_Paid)) +
  geom_bar(stat = "identity", aes(fill = -mean_Regular_Gross_Paid), width = 0.5) +
  labs(x = NULL, y = "$ Regular Gross Paid",title = "Highest Paid Employees",
       subtitle = "Highest Annual Paid") +
  theme_minimal() +
  theme(axis.text.x = element_text(size = 10),
        axis.text.y = element_text(size = 10, hjust = 0),  # Set hjust for alignment
        axis.ticks.y = NULL,
        plot.margin = margin(5, 5, 50, 5, "pt"),
        plot.title = element_text(size = 20,hjust = 0.5),
        plot.subtitle = element_text(size = 12,hjust = 0.5),
        legend.position = "none") +
        scale_y_continuous(labels = scales::comma) +  # Specify tick positions+
  coord_flip()

options(repr.plot.width = 10, repr.plot.height = 5)
```



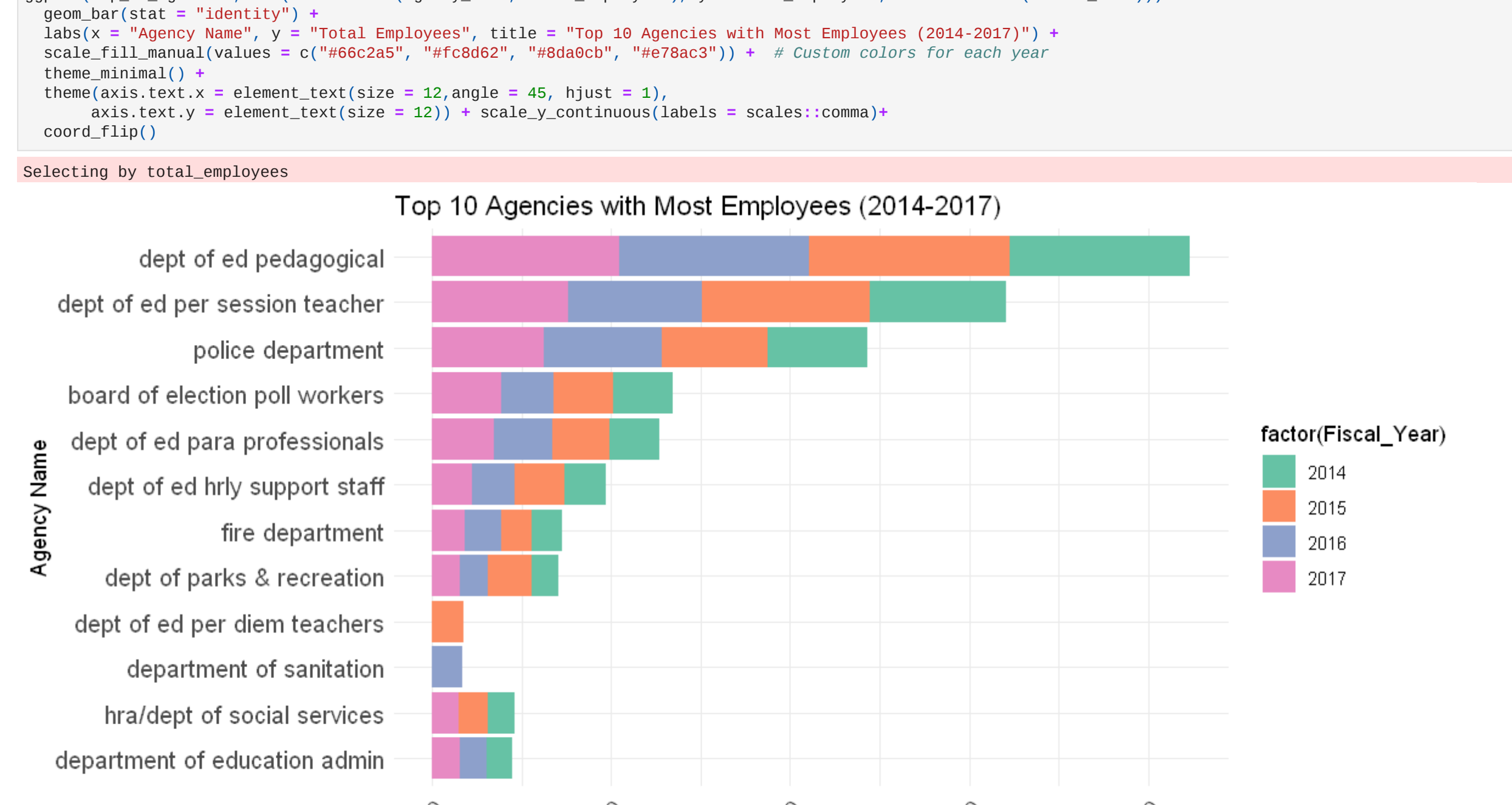
Top 10 Agencies With Most Employees

```r
In [11]: # Filter data for Fiscal_Year = 2014
data_2014 <- df[df$Fiscal_Year == 2014, ]

# Group data by Agency_Name and count employees
agency_employee_counts <- data_2014 %>%
  group_by(Agency_Name) %>%
  summarize(Employee_Count = n()) %>%
  arrange(desc(Employee_Count)) %>%
  top_n(10, wt = Employee_Count)

# Plot the top 10 agencies
ggplot(agency_employee_counts, aes(x = reorder(Agency_Name, Employee_Count), y = Employee_Count)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(x = "Agency Name", y = "Number of Employees", title = "Top 10 Agencies by Employee Count in 2014") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) + coord_flip()
```



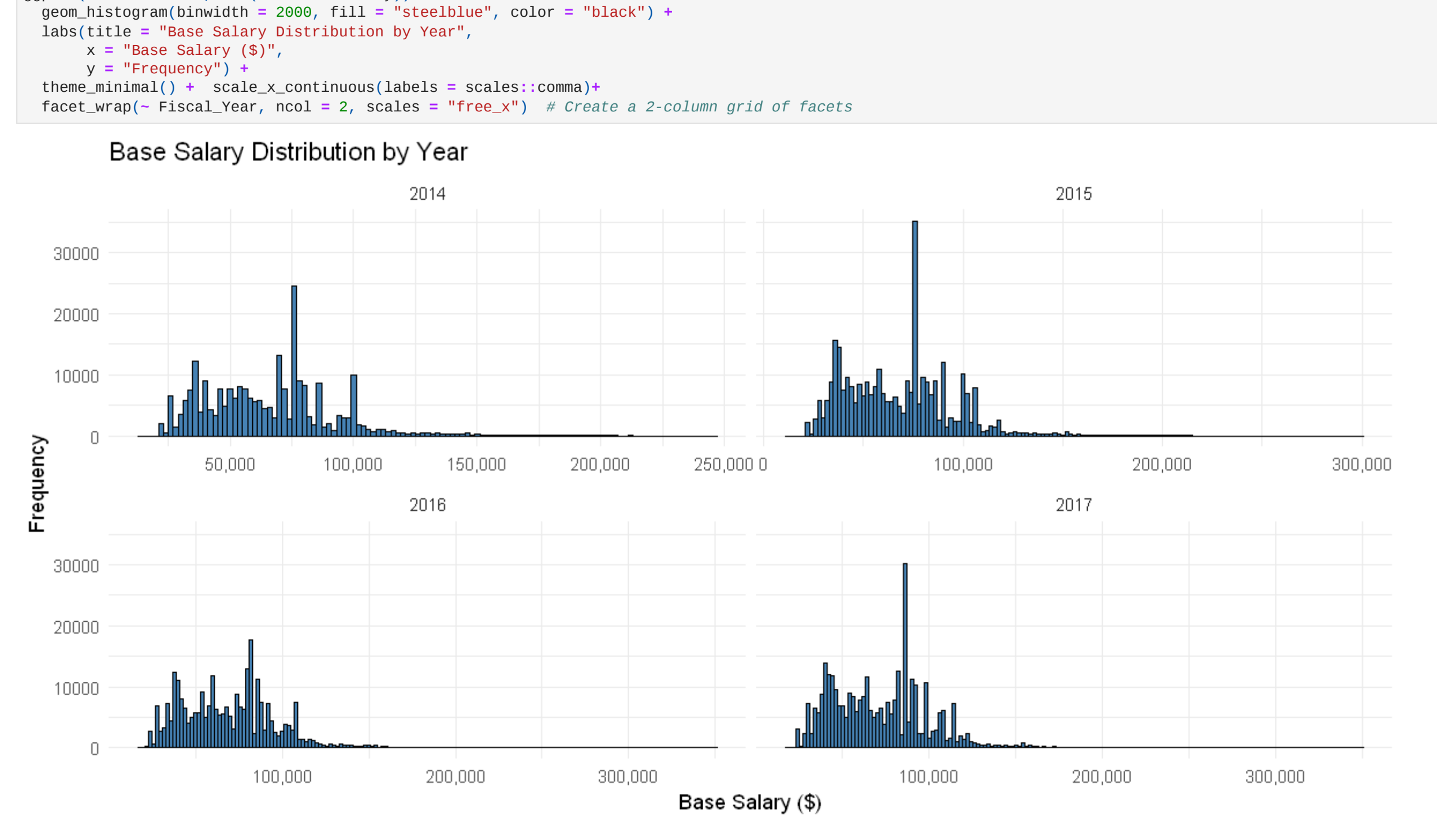

```r
In [12]: # Clean up agency names by removing extra spaces and converting to lowercase
df_clean <- df %>%
  mutate(Agency_Name = tolower(trimws(Agency_Name)))

# Group by cleaned agency name and fiscal year, calculate total number of employees
agency_employee_count <- df_clean %>%
  group_by(Agency_Name, Fiscal_Year) %>%
  summarize(total_employees = n()) %>%
  arrange(Fiscal_Year, desc(total_employees))

# Select the top 10 agencies for each fiscal year
top_10_agencies <- agency_employee_count %>%
  group_by(Fiscal_Year) %>%
  top_n(10)

# Create a bar chart
ggplot(top_10_agencies, aes(x = reorder(Agency_Name, total_employees), y = total_employees, fill = factor(Fiscal_Year))) +
  geom_bar(stat = "identity") +
  labs(x = "Agency Name", y = "Total Employees", title = "Top 10 Agencies with Most Employees (2014-2017)") +
  scale_fill_manual(values = c("#66c2a5", "#fc8d62", "#8da0cb", "#e78ac3")) +  # Custom colors for each year
  theme_minimal() +
  theme(axis.text.x = element_text(size = 12,angle = 45, hjust = 1),
        axis.text.y = element_text(size = 12)) + scale_y_continuous(labels = scales::comma)+
  coord_flip()
```

```
Selecting by total_employees
```



Base Salary Distribution (Salary > $10,000) in each year

```r
In [123…: # Filter data for each year and annual salary > 10,000
filtered_data <- df[df$Pay_Basis == " per Annum" & df$Base_Salary > 10000, ]

# Create a facetted histogram
ggplot(filtered_data, aes(x = Base_Salary)) +
  geom_histogram(binwidth = 2000, fill = "steelblue", color = "black") +
  labs(title = "Base Salary Distribution by Year",
       x = "Base Salary ($)",
       y = "Frequency") +
  theme_minimal() +  scale_x_continuous(labels = scales::comma)+
  facet_wrap(~ Fiscal_Year, ncol = 2, scales = "free_x")  # Create a 2-column grid of facets
```

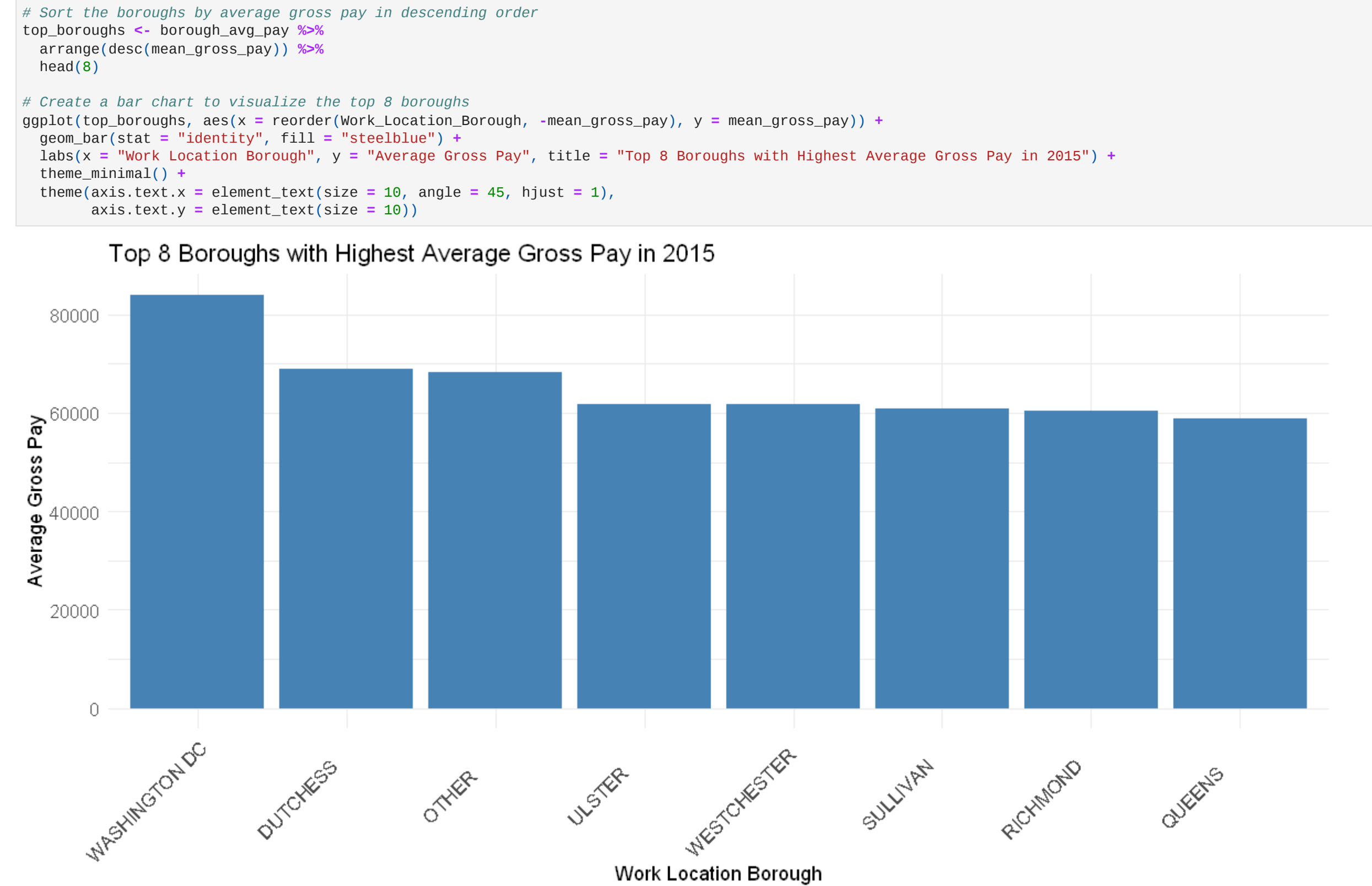


Top Boroughs with Highest Average Pay in each year

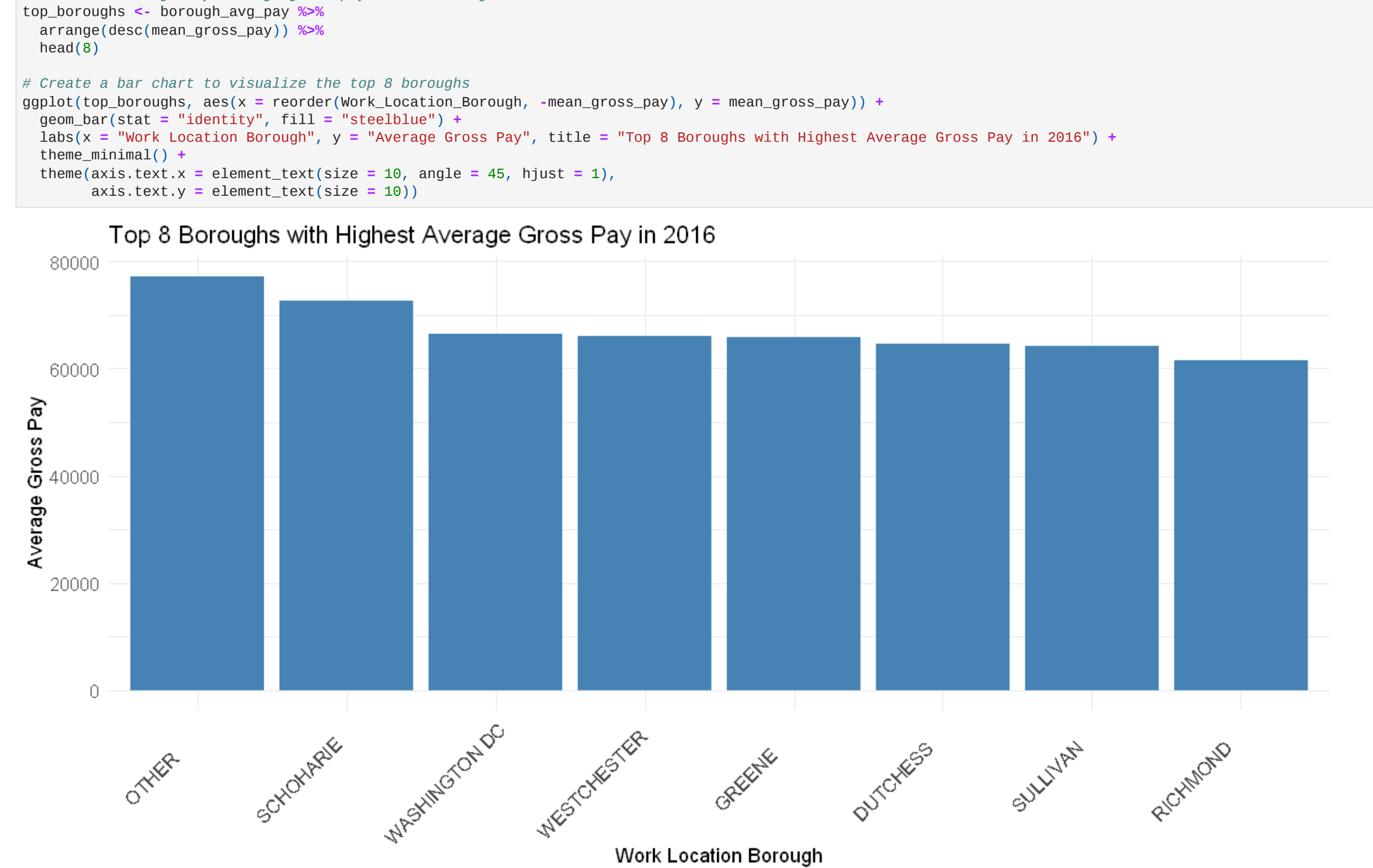
```r
In [103…: df_2015 <- df[df$Fiscal_Year == 2015, ]
df_2015 <- df_2015 %>%
  filter(`Pay_Basis` == " per Annum" & `Base_Salary` > 20000)

# Convert Work_Location_Borough to character type
df_2015$Work_Location_Borough <- as.character(df_2015$Work_Location_Borough)
df_2015 <- df_2015 %>%
  filter(Work_Location_Borough != "")

unique_boroughs <- unique(df_2015$Work_Location_Borough)
print(unique_boroughs)
```

```
 [1] "MANHATTAN    " "BROOKLYN     " "QUEENS       " "BRONX        "
 [5] "RICHMOND     " "ALBANY       " "OTHER        " "DELAWARE     "
 [9] "SULLIVAN     " "NASSAU       " "ORANGE       " "WESTCHESTER  "
[13] "ULSTER       " "PUTNAM       " "DUTCHESS     " "SCHOHARIE    "
[17] "GREENE       " "WASHINGTON DC"
```

```r
In [105…: df_2015 <- df[df$Fiscal_Year == 2015, ]
df_2015 <- df_2015 %>%
  filter(`Pay_Basis` == " per Annum" & `Base_Salary` > 20000)

# Convert Work_Location_Borough to character type
df_2015$Work_Location_Borough <- as.character(df_2015$Work_Location_Borough)
df_2015 <- df_2015 %>%
  filter(Work_Location_Borough != "")

# Calculate the average gross pay for each borough
borough_avg_pay <- df_2015 %>%
  group_by(Work_Location_Borough) %>%
  summarize(mean_gross_pay = mean(Regular_Gross_Paid))

# Sort the boroughs by average gross pay in descending order
top_boroughs <- borough_avg_pay %>%
  arrange(desc(mean_gross_pay)) %>%
  head(8)

# Create a bar chart to visualize the top 8 boroughs
ggplot(top_boroughs, aes(x = reorder(Work_Location_Borough, -mean_gross_pay), y = mean_gross_pay)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(x = "Work Location Borough", y = "Average Gross Pay", title = "Top 8 Boroughs with Highest Average Gross Pay in 2015") +
  theme_minimal() +
  theme(axis.text.x = element_text(size = 10, angle = 45, hjust = 1),
        axis.text.y = element_text(size = 10))
```




```r
In [107…: df_2016 <- df[df$Fiscal_Year == 2016, ]
df_2016 <- df_2016 %>%
  filter(`Pay_Basis` == " per Annum" & `Base_Salary` > 20000)

# Convert Work_Location_Borough to character type
df_2016$Work_Location_Borough <- as.character(df_2016$Work_Location_Borough)
df_2016 <- df_2016 %>%
  filter(Work_Location_Borough != "")

# Calculate the average gross pay for each borough
borough_avg_pay <- df_2016 %>%
  group_by(Work_Location_Borough) %>%
  summarize(mean_gross_pay = mean(Regular_Gross_Paid))

# Sort the boroughs by average gross pay in descending order
top_boroughs <- borough_avg_pay %>%
  arrange(desc(mean_gross_pay)) %>%
  head(8)

# Create a bar chart to visualize the top 8 boroughs
ggplot(top_boroughs, aes(x = reorder(Work_Location_Borough, -mean_gross_pay), y = mean_gross_pay)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(x = "Work Location Borough", y = "Average Gross Pay", title = "Top 8 Boroughs with Highest Average Gross Pay in 2016") +
  theme_minimal() +
  theme(axis.text.x = element_text(size = 10, angle = 45, hjust = 1),
        axis.text.y = element_text(size = 10))
```




```r
In [108…: df_2017 <- df[df$Fiscal_Year == 2017, ]
df_2017 <- df_2017 %>%
  filter(`Pay_Basis` == " per Annum" & `Base_Salary` > 20000)

# Convert Work_Location_Borough to character type
df_2017$Work_Location_Borough <- as.character(df_2017$Work_Location_Borough)
df_2017 <- df_2017 %>%
  filter(Work_Location_Borough != "")

# Calculate the average gross pay for each borough
borough_avg_pay <- df_2017 %>%
  group_by(Work_Location_Borough) %>%
  summarize(mean_gross_pay = mean(Regular_Gross_Paid))

# Sort the boroughs by average gross pay in descending order
top_boroughs <- borough_avg_pay %>%
  arrange(desc(mean_gross_pay)) %>%
  head(8)

# Create a bar chart to visualize the top 8 boroughs
ggplot(top_boroughs, aes(x = reorder(Work_Location_Borough, -mean_gross_pay), y = mean_gross_pay)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(x = "Work Location Borough", y = "Average Gross Pay", title = "Top 8 Boroughs with Highest Average Gross Pay in 2017") +
  theme_minimal() +
  theme(axis.text.x = element_text(size = 10, angle = 45, hjust = 1),
        axis.text.y = element_text(size = 10))
```